$\mathbf{VTC}^0$: A Second-Order Theory for $\mathbf{TC}^0$

by

Phuong Nguyen

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

$\mathbf{VTC}^0$: A Second-Order Theory for $\mathbf{TC}^0$

Phuong Nguyen

Master of Science

Graduate Department of Computer Science

University of Toronto

2004

We introduce a finitely axiomatizable second-order theory $\mathbf{VTC}^0$ and show that it characterizes precisely the class uniform $\mathbf{TC}^0$. It is simply the theory $\mathbf{V}^0$ [12] together with the axiom *NUMONES*, which states the existence of a "counting array" Y for any string X: the ith row of Y contains only the number of 1 bits upto (excluding) bit i of X. First, we introduce the notion of "strong Δ_1^B-definability" for relations in a theory, and use the recursive properties of $\mathbf{TC}^0$ relations (rather than functions) to show that $\mathbf{TC}^0$ relations are strongly Δ_1^B-definable, and $\mathbf{TC}^0$ functions are Σ_1^B-definable in $\mathbf{VTC}^0$. Then, we generalize the Witnessing Theorem for $\mathbf{V}^0$ [12], and obtain the witnessing theorem for $\mathbf{VTC}^0$ from this general result: $\exists\Sigma_{0+}^B(\Sigma_1^B)$ theorems of $\mathbf{VTC}^0$ can be witnessed by $\mathbf{TC}^0$ functions (here, $\Sigma_{0+}^B(\Sigma_1^B)$ formulas are those obtained from Σ_1^B formulas using $\wedge, \vee$ and bounded number quantifications). Finally, we show that $\mathbf{VTC}^0$ is RSUV isomorphic to the first-order theory Δ_1^b-$\mathbf{CR}$, which has been claimed the "minimal" theory for $\mathbf{TC}^0$ [20]. This isomorphism shows that $\mathbf{VTC}^0$ admits the $\Sigma_{0+}^B(\Delta_1^B)$ comprehension rule. Hence, in $\mathbf{VTC}^0$, strong Δ_1^B-definability and the usual Δ_1^B-definability coincide. It also follows that $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$, for some i. This answers affirmatively an open question from [20].

Acknowledgements

It is a great honour to work under the supervision of Professor Cook and Professor Urquhart. I would like to thank especially Professor Cook for his guide and very helpful discussions. I would like to thank Neil Thapen for many useful conversations. The financial support for this thesis is from The Department of Computer Science, University of Toronto.

Contents

Chapter 1

Introduction

Non-uniform $\mathbf{TC}^0$ is the class of languages (or sometimes functions) computable by families of polynomial-size, constant-depth circuits with majority gates. Uniform $\mathbf{TC}^0$ is defined similarly with the restriction that the families of circuits are uniform. The commonly accepted notion of uniformity for $\mathbf{TC}^0$ is $\mathbf{AC}^0$-uniform, or equivalently **FO**-uniform [17, page 79]. We will simply use $\mathbf{TC}^0$ for **FO**-uniform $\mathbf{TC}^0$ (as a class of languages, or relations, the class of functions will be called $\mathbf{FTC}^0$, precise definitions will be given in the next chapter). Note that threshold gates (i.e., gates which count the number of inputs which are 1) can be simulated using majority gates. Therefore, there is an equivalent definition of $\mathbf{TC}^0$ using threshold circuits, hence the name $\mathbf{TC}^0$ (where 0 indicates $\log^0$-depth, i.e., constant-depth).

Even though it is defined in a restricted way, $\mathbf{TC}^0$ has not been separated from "big" classes, such as **NP**. In fact, an increasing number of important functions and decision problems have been shown in $\mathbf{TC}^0$. For example, integer multiplication, iterated integer multiplication, integer division as well as sorting are indeed complete for $\mathbf{TC}^0$ under **FO** reduction [9, 6, 15, 3]. (Note that since $\mathbf{TC}^0 \subseteq \mathbf{L}$, $\mathbf{TC}^0 \subset \mathbf{DSPACE}(f(n))$, for any function f that grows faster than $\log n$.)

In studying complexity classes, the logical theories play an important role. For example, the hierarchy $\mathbf{S}$ of first-order theories $\mathbf{S}_2^1 \subseteq \mathbf{T}_2^1 \subseteq \mathbf{S}_2^2 \subseteq \ldots$, introduced by Buss [4], characterizes precisely $\mathbf{PH}$, the polynomial hierarchy. It has been shown that $\mathbf{PH}$ collapses if and only if $\mathbf{S}$ collapses, or equivalently $\mathbf{S}$ is finitely axiomatizable [18]. In the case of $\mathbf{TC}^0$, a number of theories have been proposed. Perhaps the most remarkable among them is the first-order theory Δ_1^b-$\mathbf{CR}$ by Johannsen and Pollett [20], who have shown (among other results) that:

$$\text{If } \Delta_1^b - \mathbf{CR} = \mathbf{S}_2^1, \text{ then } \mathbf{NP} \text{ is contained in non-uniform } \mathbf{TC}^0.$$

In this thesis, we introduce the second-order theory $\mathbf{VTC}^0$, and show that it characterizes $\mathbf{TC}^0$. We will also show that $\mathbf{VTC}^0$ is RSUV isomorphic to Δ_1^b-$\mathbf{CR}$. From this, the result by Johannsen and Pollett can be translated directly into second-order setting. It also helps to close an open question in [20], i.e., $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$, for some i.

1.1 Uniformity

The common notion of uniformity for $\mathbf{TC}^0$ is $\mathbf{FO}$-uniform. By Theorem 5.22 in [17, page 82], this is equivalent to $\mathbf{AC}^0$-uniformity. Essentially, a family of circuits is $\mathbf{FO}$-uniform if they can be described by a set of formulas in the first-order vocabulary which contains symbols specifying the connections between gates, and the types of the gates in the circuits.

Although $\mathbf{FO}$-uniformity is rather weak, basic properties of $\mathbf{TC}^0$ can be shown by manipulating the circuits. This may require showing various properties of class of uniform families of circuits, e.g., it is closed under some operations, such as composition, negation, etc. Since these often seems "trivial", formal proofs are usually omitted.

We will give formal proofs for various properties of $\mathbf{TC}^0$ and $\mathbf{FTC}^0$, based on a result by Barrington, Immerman and Straubing [2] which shows that $\mathbf{TC}^0$ is exactly the class of languages definable in $\mathbf{FO}(COUNT)$ (i.e., first-order logic with the counting quantifier). By an obvious coding scheme, we can define the class of $\mathbf{TC}^0$ relations, which, without ambiguity, is also called $\mathbf{TC}^0$. Then, the $\mathbf{TC}^0$ functions are those whose bitgraphs are in $\mathbf{TC}^0$ and whose value are properly bounded. Now, the properties of functions in $\mathbf{FTC}^0$ can be proved using properties of their bitgraphs, while the properties of $\mathbf{TC}^0$ relations can be carried out formally using Immerman's characterization of $\mathbf{TC}^0$.

An application of Barrington, Immerman and Straubing's result is that we can prove various properties of $\mathbf{VTC}^0$ by induction on the structure of the $\mathbf{FO}(COUNT)$ sentences that define $\mathbf{TC}^0$ languages. However, we find that it is not convenient to prove results using induction on the structure of *sentences*. In fact, we will translate $\mathbf{FO}(COUNT)$ sentences into formulas in second-order logic with counting quantifier, which represent the same languages as those defined by the $\mathbf{FO}(COUNT)$ sentences. Then, we use induction on the structure of *formulas* (in the new logic) to prove these properties. The translation is straightforward, and similar to the case where the counting quantifier is not used, as discussed in [12].

1.2 Theories for $\mathbf{TC}^0$

In general, theories are developed for complexity classes so that reasoning in the theories reflect the computational power of the classes. In particular, functions and relations in the complexity class are definable in the theories by some classes of formulas. For example, the class of Σ_1^b-definable functions in $\mathbf{S}_2^1$ is exactly $\mathbf{FP}$, and the class of Δ_1^b-definable predicates in $\mathbf{S}_2^1$ is exactly $\mathbf{P}$ [4, 8]. Similar results connecting the theory $\mathbf{T}_2^1$ and the class $\mathbf{PLS}$ (polynomial local search) have been shown [22].

While first-order logic has proved successful in characterizing complexity classes, second-order (or rather two-sorted) logic provides a more elegant presentation. In the case of $\mathbf{TC}^0$, second-order logic has one more advantage over first-order logic. It is known that the multiplication function is computable in $\mathbf{TC}^0$ [3]. In fact, its graph is complete for $\mathbf{TC}^0$ under $\mathbf{AC}^0$ reduction [6]. Thus, conceivably, any $\mathbf{TC}^0$ function can be definable from multiplication. However, first-order theories naturally contain defining axioms for multiplication, in addition to other axioms or rules. This might be the reason explaining why the first-order theories for $\mathbf{TC}^0$ (discussed below) often have "unnatural" syntactical restrictions.

1.2.1 First-Order Theories for $\mathbf{TC}^0$

In [11], Clote and Takeuti introduce the notion of *essentially sharply bounded* (esb) formulas in a theory $\mathcal{T}$. Then they propose the first-order theory $\mathbf{TTC}^0$, and show that a function belongs to $\mathbf{TC}^0$ if and only if it is esb-definable in $\mathbf{TTC}^0$. However, the definition of esb formulas in a theory is already complicated, and the structure of esb formulas in a theory like $\mathbf{TTC}^0$ seems understandable only by examining the class of esb-definable functions of $\mathbf{TTC}^0$, i.e., the class of $\mathbf{TC}^0$ functions, which is itself the subject of investigation. Furthermore, $\mathbf{TTC}^0$ is defined using complicated axioms, such as $ep\Sigma_1^b$-*BLIND*, which seems not practical. As a result, there have not been many application of the theory $\mathbf{TTC}^0$.

In [19], Johannsen introduce the first-order theory $\overline{\mathbf{R}}^0$, and show that it captures exactly $\mathbf{TC}^0$: the class of $\mathbf{TC}^0$ functions is exactly the class of functions Σ_1^b-definable in $\overline{\mathbf{R}}^0$. In [21], Johannsen and Pollett introduce a hierarchy $\{\mathbf{C}_k^0\}_{k\geq 1}$ of first-order theories, which characterizes the *counting hierarchy*. They show that $\mathbf{C}_2^0$ also captures $\mathbf{TC}^0$. Although $\overline{\mathbf{R}}^0$ and $\mathbf{C}_2^0$ characterize precisely $\mathbf{TC}^0$, they seem too "strong" for $\mathbf{TC}^0$: they might have proper sub-theories which also characterize $\mathbf{TC}^0$ in the same way. In fact, the first-order theory Δ_1^b-$\mathbf{CR}$ introduced by Johannsen and Pollett also captures $\mathbf{TC}^0$, and can be seen as a minimal theory for $\mathbf{TC}^0$ [20]. It is easy to see that Δ_1^b-$\mathbf{CR}$ is a sub-theory of both $\overline{\mathbf{R}}^0$ and $\mathbf{C}_2^0$. Moreover, Cook and Thapen [14] show that $\mathbf{PV}$ does not prove the Σ_0^b replacement axiom scheme, unless RSA can be cracked in polynomial time. The same arguments can carried over for Δ_1^b-$\mathbf{CR}$, and thus Δ_1^b-$\mathbf{CR}$ is very likely a proper sub-theory of both $\overline{\mathbf{R}}^0$ and $\mathbf{C}_2^0$, since they both contain the Σ_0^b replacement axiom scheme.

The theory Δ_1^b-**CR** is defined using a set of axioms (i.e., *BASIC* together with *Open-LIND*), and the Δ_1^b *comprehension rule*. Essentially, this rule gives an inductive definition of the theory Δ_1^b-**CR**: it can be seen as built in an infinite number of stages, starting with *BASIC* and *Open-LIND*. At each stage, it is expanded by taking the closure under entailment and by adding the conclusions of all instances of the Δ_1^b comprehension rule whose top sequent is in the current theory. Note that the result by Cook and Thapen [14] suggests that this inference rule may not be equivalent to the corresponding Δ_1^b *comprehension axiom*. In fact, there has not been a nice axiomatization of Δ_1^b-**CR**. In this thesis, we will introduce the second-order theory $\mathbf{VTC}^0$, which is finitely axiomatizable, captures $\mathbf{TC}^0$, and is RSUV isomorphic to Δ_1^b-**CR**. In the next part, we will discuss the presence of second-order theories for $\mathbf{TC}^0$ in the literature.

1.2.2 Second-Order Theories for $\mathbf{TC}^0$

In [21], the first-order theories $\mathbf{C}^0_{k+1}$ $(k \geq 1)$ have been shown RSUV isomorphic to the second-order theories $\mathbf{D}^0_k$. Thus, $\mathbf{D}^0_1$ can be seen as a theory for $\mathbf{TC}^0$. As already discussed above for the theory $\mathbf{C}^0_2$, even though it correctly characterizes $\mathbf{TC}^0$, $\mathbf{D}^0_1$ might be unnecessarily strong for $\mathbf{TC}^0$.

In [23], Krajíček searches for theories corresponding to polynomial-size Frege proofs. He introduces the theory $(I\Sigma_1^{1,b})^{count}$ and notes that it corresponds to the extension **FC** of constant-depth Frege proof system. He also poses the question of whether **FC** p-simulates Frege proof systems. It turns out that our theory $\mathbf{VTC}^0$ is the same as $(I\Sigma_1^{1,b})^{count}$. Consequently, Krajíček's question is related to the question of whether $\mathbf{TC}^0 = \mathbf{NC}^1$.

As can be seen, there have not been many serious attempts to develop second-order theories for $\mathbf{TC}^0$. One reason might be that first-order logic has been well studied, with available tools such as compactness theorem, Herbrand Theorem, etc. Another reason might be that the syntax of second-order bounded arithmetic had been quite heavy before its elegant presentation in [26, 12]. It seems that for "low" complexity classes such as $\mathbf{TC}^0$, second-order logic is a more natural choice, since we are not forced to define multiplication for second-order objects, which is already complete for $\mathbf{TC}^0$.

1.2.3 Witnessing in $\mathbf{VTC}^0$ and Defining $\mathbf{TC}^0$ Functions and Relations

In order to show that a function class is definable in a theory (using some class of formulas), one often needs the recursive characteristic of the function class. For example, the class of recursive functions is the closure of a set of initial functions under composition, primitive recursion and minimization. The same techniques have been employed for the class of $\mathbf{TC}^0$ functions. A recursive characteristic (also called function algebra) of $\mathbf{TC}^0$ functions is that they are obtained from $0, I, s_0, s_1, |\ |, Bit, \cdot, \#$ under composition and *concatenation recursion on notation* (here, I is the collection of all projection functions; s_0, s_1 are the binary successor functions; $|\ |$ returns the length of the binary representation of a number; Bit returns the bit at a specific position in the binary representation of a number; and $\#$ is the smash function: $x\#y = 2^{|x||y|}$) [11, 10].

We will show that the $\mathbf{TC}^0$ functions are exactly those Σ_1^B-definable in $\mathbf{VTC}^0$, and that the $\mathbf{TC}^0$ relations are exactly those Δ_1^B-definable in $\mathbf{VTC}^0$. First, we will prove a recursive characteristic of $\mathbf{TC}^0$ *relations*: they are precisely the closure of $\mathbf{AC}^0$ relations under Boolean and *counting* operations. This comes naturally from the definition of $\mathbf{TC}^0$ (i.e., using threshold circuits). Using this property, we show that the $\mathbf{TC}^0$ relations are *strongly* Δ_1^B*-definable* in $\mathbf{VTC}^0$. Then we show a general result relating Σ_1^B-definable functions and strongly Δ_1^B-definable relations: a function is Σ_1^B-definable in a bounded theory $\mathcal{T}$ if and only if it is bounded, and its bitgraph is strongly Δ_1^B-definable in $\mathcal{T}$.

The *witnessing theorem* for $\mathbf{VTC}^0$, which states that the Σ_1^B theorems of $\mathbf{VTC}^0$ are witnessed by $\mathbf{TC}^0$ functions, follows from a more general phenomenon: it is straightforward to witness the Σ_1^B theorems of a Σ_0^B theory $\mathcal{T}$ if the vocabulary of $\mathcal{T}$ is "rich" enough. We will show that the vocabulary of $\mathbf{VTC}^0$ can be "enriched" by adding the symbols for $\mathbf{TC}^0$ functions, together with their Σ_0^B defining axioms. At the same time, we obtain a Σ_0^B theory $\overline{\mathbf{VTC}}^0$ which is conservative over $\mathbf{VTC}^0$. Witnessing theorem for $\mathbf{VTC}^0$ will then follow by applying the general result for $\overline{\mathbf{VTC}}^0$.

1.3 RSUV Isomorphism

The equivalence between first-order and second-order theory is made precise by the notion of RSUV isomorphism [25] (see also [24]). A first-order theory $\mathcal{T}_1$ and a second-order theory $\mathcal{T}_2$ are RSUV isomorphic if each can be interpreted in the other. Interpreting $\mathcal{T}_2$ in $\mathcal{T}_1$ involves mapping $\mathcal{T}_2$'s strings (i.e., second-order objects) to $\mathcal{T}_1$'s numbers, $\mathcal{T}_2$'s numbers to $\mathcal{T}_1$'s "small numbers", and checking that the mapped axioms hold in the first-order theory. Interpreting $\mathcal{T}_1$ in $\mathcal{T}_2$ is the reverse process. While the first direction is often straightforward, the second direction is sometimes less obvious. One of our main obstacles in proving RSUV isomorphism between $\mathbf{VTC}^0$ and Δ_1^b-$\mathbf{CR}$ is defining (string) multiplication and proving its properties.

Defining string multiplication in $\mathbf{V}^1$ (while proving that $\mathbf{V}^1$ and $\mathbf{S}_2^1$ are RSUV isomorphic [24, 25, 16]) is possible by using induction on Σ_1^B formulas (i.e., formulas of the form $\exists \overline{X} \leq b\phi$, where ϕ is a bounded formula with no string quantifier). This induction scheme might not be available in $\mathbf{VTC}^0$, hence defining multiplication and proving its properties in weak theories such as $\mathbf{VTC}^0$ does not appear straightforward to us. Here, we have to formalize a number of "non-trivial" concepts in $\mathbf{VTC}^0$. We also adopt the approach in [16], where multiplication is defined in a symmetric way, in order to simplify the isomorphism proof. Note that in [21], the theories $\mathbf{C}_{k+1}^0$ and $\mathbf{D}_k^0$ are shown RSUV isomorphic. In particular, $\mathbf{C}_2^0$ and $\mathbf{D}_1^0$ are RSUV isomorphic. Correspondence with the authors of [21] shows that their proof of the RSUV isomorphism uses Δ_1^B *comprehension axiom*, which might not be provable in $\mathbf{VTC}^0$, as we have discussed.

1.4 Organization

In the next chapter, we will formally define $\mathbf{TC}^0$ and $\mathbf{FTC}^0$. We will then translate results in descriptive complexity into a second-order logic with the counting quantifier. This enables us to prove properties of $\mathbf{TC}^0$ and $\mathbf{FTC}^0$. In Chapter 3, we define the theory $\mathbf{VTC}^0$, and prove that it captures exactly $\mathbf{TC}^0$. Then, in Chapter 4 we will show that $\mathbf{VTC}^0$ and Δ_1^b-$\mathbf{CR}$ are RSUV isomorphic. Finally, Chapter 5 concludes the thesis and discuss possible future research directions.

Chapter 2

The Class FO-Uniform $\mathbf{TC}^0$

In this chapter, we will formally define the classes **FO**-uniform $\mathbf{TC}^0$ and $\mathbf{FTC}^0$. Non-uniform $\mathbf{TC}^0$ (respectively $\mathbf{FTC}^0$) is the class of languages (respectively functions) that are computable using families of polynomial-size, constant-depth threshold circuits. Uniform $\mathbf{TC}^0$ and $\mathbf{FTC}^0$ are defined similarly, with the restriction that the families of circuits are uniform. From this definition, one is able to prove their various properties. These proofs often involves manipulating uniform families of circuits (e.g., composing). While this is possible, even with a very weak notion of uniformity like **FO**-uniformity, the formal proofs may be tedious. Here, we will confine the issue of uniformity to only the definition of $\mathbf{TC}^0$. The class $\mathbf{FTC}^0$ will be defined in terms of $\mathbf{TC}^0$, and its properties will be proved using the properties of $\mathbf{TC}^0$. By results from descriptive complexity [17], we will be able to handle $\mathbf{TC}^0$ using its logical characteristics.

Since we will be dealing with both numbers and strings (e.g., inputs, outputs of a functions computed by a family of circuits), it is convenient to work in second-order logics, where the second-order objects can be interpreted as binary strings.[1] In this chapter only, our second-order logic is augmented with the counting quantifier. This is a useful tool for proving basic properties of the $\mathbf{TC}^0$ relations. In fact, moving from descriptive complexity

[1] the second-order logic that we use here can also be regarded as a two-sorted logic

to this augmented second-order logic is almost transparent, while manipulating relations on numbers and strings in this logic is much easier than in descriptive complexity.

In the following sections, first, we recall concepts from descriptive complexity. Then, we define **FO**-uniform $\mathbf{TC}^0$ as a class of languages. The definition is naturally extended to class of relations on binary strings. Finally, we will prove some important properties of $\mathbf{TC}^0$ and $\mathbf{FTC}^0$.

2.1 Descriptive Complexity

In this part we will be dealing only with *relational vocabularies* of first-order logic, i.e., vocabularies which do not contain function symbols of arities > 0. For a first-order language $\mathcal{L}$, let STRUCT[$\mathcal{L}$] denote the class of all $\mathcal{L}$ structures. For a structure $\mathscr{A}$, let $\|\mathscr{A}\|$ denote the universe of $\mathscr{A}$. For each (non-empty) binary string X, let $|X|$ denote the length of X. First, consider the first-order vocabulary $\mathcal{L}_{\mathcal{FO}}$, where

$$\mathcal{L}_{\mathcal{FO}} = [0, 1, max; =, \leq, BIT, SUC, Z].$$

For each non-empty binary string X, denote by $\mathscr{S}_X$ the $\mathcal{L}_{\mathcal{FO}}$-structure where

$$\|\mathscr{S}_X\| = \{0, \dots, |X|-1\}, \quad max^{\mathscr{S}_X} = |X|-1, \quad BIT(i,x) \text{ iff the } i\text{th bit of x is } 1,$$

$$Z^{\mathscr{S}_X} = \{i < |X| : \text{ the } i \text{ bit of } X \text{ is } 1\},$$

and other symbols are interpreted naturally. Then, each $\mathcal{L}_{\mathcal{FO}}$-sentence φ defines a language $L(\varphi)$ in the following way: $L(\varphi)$ is a set of binary strings whose associated structures satisfy φ. Formally,

$$L(\varphi) = \{X \in \{0,1\}^+ : \mathscr{S}_X \models \varphi\}.$$

The complexity class **FO** is the class of languages definable by some first-order sentences,

$$\mathbf{FO} = \{L : L = L(\varphi) \text{ for some } \mathcal{L}_{\mathcal{FO}}\text{-sentence } \varphi\}.$$

It has been shown [17] that this class is the same as a version of uniform $\mathbf{AC}^0$.

Also in [2, 17], an extension of first-order logic has been considered, in which a new quantifier (namely, the *counting* quantifier $\exists i\ x$) is added. Its meaning is that $\exists i\ y\varphi(y)$ is true if and only if there are exactly i values of y which satisfy φ. More precisely, by $\mathcal{L}_{\mathcal{FO},COUNT}$ formulas we mean the formulas built from $\mathcal{L}_{\mathcal{FO}}$ together with the counting quantifier in the usual way. The truth value of an $\mathcal{L}_{\mathcal{FO},COUNT}$ formula in a structure is defined in the same manner as usually. Then, analogous to **FO**, let $\mathbf{FO}(COUNT)$ be the class of languages definable by some $\mathcal{L}_{\mathcal{FO},COUNT}$ sentences:

$$\mathbf{FO}(COUNT) = \{L : L = L(\varphi) \text{ for some } \mathcal{L}_{\mathcal{FO},COUNT}\text{-sentence } \varphi\}.$$

This class has shown to be equal to **FO**-uniform $\mathbf{TC}^0$, whose definition is given below.

An uniform version of $\mathbf{TC}^0$ is a class of languages accepted by some uniform families of polynomial-size constant-depth circuits with threshold gates (where all the gates have unbounded fan-in). A family of circuits is **FO**-uniform if the members of the family can be described using some first-order formulas. First, description of a circuit includes a numbering of the gates, a specification of the type of each gate, and a specification of the connections between gates. Hence, a threshold circuit can be seen as a structure in the vocabulary

$$\mathcal{L}_{tc} = [0, 1, max, r; =, \leq, BIT, SUC, E^2, G^1_\wedge, G^1_\vee, G^1_\neg, G^2_{tc}, I^1],$$

where r specifies the root (or output) of the circuit, E specifies the connections, $G_\wedge, G_\vee, G_\neg, G_{tc}$ specify the types of the gates ($G_{tc}(g, v)$ means that v is the threshold value for gate g), and I specifies gates that contain constant 1.

Next, uniformly describing a family $\{\mathcal{C}_n\}$ of circuits by a set Φ of formulas essentially means that for each input length n, $\mathcal{C}_n$ can be defined by Φ in a $\mathcal{L}_{\mathcal{FO}}$-structure of size n. Hence, Φ can be seen as specifying a mapping from $\mathcal{L}_{\mathcal{FO}}$-structures to $\mathcal{L}_{tc}$-structures. This kind of mapping is generalized by the notion *first-order query.* Given two vocabularies $\mathcal{L}_1$ and $\mathcal{L}_2$. A first-order query is a mapping q from STRUCT$[\mathcal{L}_1]$ to STRUCT$[\mathcal{L}_2]$ such that for an $\mathcal{L}_1$ structure $\mathscr{A}$, the universe, constants and predicates of $q(\mathscr{A})$ are definable by some $\mathcal{L}_1$ formulas. A formal definition from [17] is as follows.

Definition 2.1 *(***First-Order Queries***) Suppose $\mathcal{L}_1$, $\mathcal{L}_2$ are two vocabularies; $\mathcal{L}_2 = [c_1, \ldots, c_l; R_1, \ldots, R_k]$, where R_i has arity a_i, for $1 \leq i \leq k$. A query from $STRUCT[\mathcal{L}_1]$ to $STRUCT[\mathcal{L}_2]$ is a mapping*

$$q : STRUCT[\mathcal{L}_1] \mapsto STRUCT[\mathcal{L}_2],$$

such that for $\mathcal{A} \in STRUCT[\mathcal{L}_1]$, $\mathscr{B} = q(\mathscr{A})$ can be specified by a number $m \in \mathbb{N}$ and $(l + k + 1)$ $\mathcal{L}_1$ formulas $\phi, \varphi_1, \ldots, \varphi_l, \psi_1, \ldots, \psi_k$ as follows:

$$\|\mathscr{B}\| = \{\bar{b} \in \|\mathscr{A}\|^m : \mathscr{A} \models \phi(\bar{b})\}, \;\; c_i^{\mathscr{B}} = \bar{b} \in \|\mathscr{A}\|^m : \mathscr{A} \models \varphi_i(\bar{b}), \;\; R_j^{\mathscr{B}} = \{\bar{b} \in \|\mathscr{A}\|^m : \mathscr{A} \models \psi_j(\bar{b})\},$$

where for each φ_i, there is an unique $\bar{b} \in \|\mathscr{A}\|^m$ such that $\varphi_i(\bar{b})$.

Now, for $n \geq 1$, let $\mathscr{S}_n$ be the structure $\mathscr{S}_X$, where X is a string consisting of n 0's. Then a sequence of threshold circuits $\{\mathcal{C}_n\}$ is **FO**-uniform if there is a query $q : \mathrm{STRUCT}[\mathcal{L}_{\mathcal{FO}}] \rightarrow \mathrm{STRUCT}[\mathcal{L}_{tc}]$, such that for all $n \in \mathbb{N}$, $q(\mathscr{S}_n) = \mathcal{C}_n$.

Definition 2.2 *(***The Class $\mathbf{TC}^0$***) Let $\mathbf{TC}^0$ denote the class of languages accepted by* **FO***-uniform families of constant-depth, polynomial-size unbounded fan-in threshold circuits.*

Theorem 2.3 *(***[2, 17]***)* **FO**$(COUNT) = \mathbf{TC}^0$*.*

Remark 2.4 *(*$\mathbf{TC}^0$ **As a Class of Relations***) We have defined $\mathbf{TC}^0$ as a class of languages. If we assume some simple encoding scheme for tuples of binary strings and numbers (as unary strings), then we can also view it as a class of relations on numbers and strings. Details of the encoding are immaterial. For example, we can use new symbols "%" to separates the string part from the number part, and "\$" to separate arguments in each part. First, we write down the unary strings for the numbers $\overline{x}$ (separated by \$),* *then the symbol %, then the binary strings* $\overline{Y}$ *(separated by $). Thus, we obtain a string of the form*

$$\$x_1\$\ldots\$x_k\%Y_1\$\ldots\$Y_l\$.$$

Then, a binary representation can be obtained by writing 00 for 0, 01 for 1, 10 for "$", and 11 for "%". Let $[\overline{x}, \overline{Y}]$ *denote the binary string encoding the tuple* $(\overline{x}, \overline{Y})$*. We say a relation* $R(\overline{x}, \overline{Y})$ *is in a complexity class if the associated language* $\{[\overline{x}, \overline{Y}] : R(\overline{x}, \overline{Y})\}$ *is in that class. Thus, from now on, we will refer to* $\mathbf{TC}^0$ *as a class of relations.*

2.2 Second-Order Logic

In this part, we will give formal definition of the second-order logic that we are using. The materials are mainly from [12], we also follow the convention set there. Consider an extension of first-order logic, where there are two sorts of variables: the number variables $x, y, z, \ldots$ and set variables $X, Y, Z, \ldots$, whose intended values are finite set of numbers. Formally, consider the vocabulary $\mathcal{L}_A^2$ which extends the vocabulary of Peano Arithmetic:

$$\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ |; \in, \leq, =^1, =^2].$$

Here, $|\ |$ is the symbol for a function from strings to numbers, the intended meaning of $|X|$ is 1 plus the largest element of X. The binary predicate $\in$ denotes set membership. We will use the abbreviation $X(t)$ for $t \in X$. The equality symbols $=^1$ and $=^2$ are for numbers and sets, respectively. We will write $=$ for both $=^1$ and $=^2$; the exact meaning will be clear from the context. The other constant, function and predicate symbols have their standard meanings.

2.2.1 Syntax and Semantics of Second-Order Logic

The syntax is similar to that of first-order logic. The differences come from the addition of second-order terms. Consider a vocabulary $\mathcal{L}$ extending $\mathcal{L}_A^2$. We will define $\mathcal{L}$ terms and $\mathcal{L}$ formulas. We will refer to them simply as terms or formulas, when the meaning is clear from the context.

Definition 2.5 *(***Terms***) The constants 0 and 1, and the number variables are number terms. Set variables are set terms. If $\overline{s}$ are number terms, $\overline{T}$ are set terms, f is a number function symbol, and G is a set function symbol, then $f(\overline{s},\overline{T})$ is a number term and $G(\overline{s},\overline{T})$ is a set term (assuming that the arities of f and G match with the length of $\overline{s}$ and $\overline{T}$).*

The $\mathcal{L}$ formulas are also defined similarly to first-order logic, with the addition of atomic formulas of the form $T(s)$ for any set term T and number term s, and set quantifications.

Definition 2.6 *(***Formulas***) Suppose that r, s are number terms, and T is a set term, then $r = s, r \leq s, T(s)$ are atomic formulas. Also, $P(\overline{s},\overline{T})$ is an atomic formula, for predicate symbol P, number terms $\overline{s}$ and set terms $\overline{T}$ (assuming the length of $\overline{s}$ and $\overline{T}$ match with the arity of P). In addition, if φ, ψ are formulas, x a number variable, and T a set variable, then $(\varphi \wedge \psi), (\varphi \vee \psi), (\neg\varphi), (\exists x\varphi), (\forall x\varphi), (\exists X\varphi)$ and $(\forall X\varphi)$ are formulas.*

The bounded formulas $\exists x \leq t\varphi, \forall x \leq t\varphi$ are defined as usual, and $\exists X \leq t\varphi, \forall X \leq t\varphi$ stand for $\exists X(|X| \leq t \wedge \varphi)$, $\forall X(|X| \leq t \supset \varphi)$, respectively. As in first-order logic, the semantics of $\mathcal{L}_A^2$ is given by structures and object assignments. Note that $=^1$ and $=^2$ are true equality relations in any structure.

Definition 2.7 *(***Second-Order Structures***) A structure $\mathcal{M}$ for $\mathcal{L}$ consists of*

- *Two non-empty sets U_1 and U_2: the universes for number and set objects, respectively;*
- *Elements, functions and relations interpreting the function and predicate symbols in U_1 and U_2.*

Note that, in particular, there are (i) elements, functions and predicate in U_1 interpreting $0, 1, +, \cdot, \leq$; (ii) function $|\ |^{\mathcal{M}} : U_2 \mapsto U_1$ interpreting $|\ |$; and (iii) binary relation $\in^{\mathcal{M}} \subseteq U_1 \times U_2$ interpreting $\in$. The truth value of a formula in a structure under an object assignment is as usual.

The standard structure $\underline{\mathbb{N}}_2$ has U_1 equals to $\mathbb{N}$; U_2 the set of finite subsets of $\mathbb{N}$; $|S|$ is 0 if $S = \emptyset$, and 1 plus the largest member of S otherwise; and the other symbols get their standard meaning.

2.2.2 Representing Relations on Numbers and Strings

First, we recall the string representations of finite subsets of $\mathbb{N}$ from [12]. Suppose S is a non-empty finite subset of $\mathbb{N}$, $|S| = n, n \geq 1$ (i.e., $n-1$ is the largest element of S). Let $w(S)$ be the binary string

$$w(S) = S(0) \ldots S(n-2),$$

where $S(i) = 1$ if $i \in S$, and $S(i) = 0$ otherwise ($w(S)$ is the empty string ϵ if $n = 1$). Then $w(S)$ can be seen as a binary representation of S. The map w is a bijection between the non-empty finite subsets of $\mathbb{N}$ and $\{0,1\}^*$. We will extend w so that $w(\emptyset) = \epsilon$ (thus, w will no longer be a bijection, but this will not create any problem). Since this mapping is often obvious, we will use "set" and "string" interchangeably.

Next, suppose that $\varphi(\overline{x}, \overline{Y})$ is a formula with all free variables indicated. Then φ represents the relation R of numbers and strings, where

$$R = \{(\overline{a}, \overline{w(S)}) : \underline{\mathbb{N}}_2 \models \varphi(\overline{a}, \overline{S})\},$$

where $\overline{w(S)}$ means $w(S_0), \ldots, w(S_n)$, and n is the length of $\overline{X}$.

Notation 2.8 *Fix a vocabulary $\mathcal{L}$ which extends $\mathcal{L}_A^2$. A formula φ is a $\Sigma_0^B(\mathcal{L})$ formula if it has only bounded number quantifiers. A $\Sigma_1^B(\mathcal{L})$ ($\Pi_1^B(\mathcal{L})$) formula is a $\Sigma_0^B(\mathcal{L})$ formula preceded by a block of bounded existential (universal) string quantifiers, i.e., quantifiers* *of the form $\exists X \leq t$ ($\forall X \leq t$). If the block contains a single quantifier, the formula is also called a single-$\Sigma_1^B(\mathcal{L})$ (single-$\Pi_1^B(\mathcal{L})$) formula. We will not mention $\mathcal{L}$ if it is clear from the context. Also, Σ_1^1 formulas are those of the form $\exists \overline{Z}\varphi(\overline{Z})$, where φ is Σ_0^B. We will also use $g\Sigma_1^B$ ($g\Pi_1^B$) formulas to refer to those which are obtained from Σ_0^B formulas using the connectives $\wedge$ and $\vee$, bounded number quantifications and bounded existential (universal) string quantification.*[2]

The Σ_0^B Representation Theorem (Page 54 [12]) states that a relation is in $\mathbf{AC}^0$ if and only if it is represented by a Σ_0^B formula. We will introduce the counting quantifier for second-order logic, and prove a similar result for $\mathbf{TC}^0$ relations.

2.2.3 Adding the Counting Quantifier to Second-Order Logic

Similar to the case of first-order logic, we can add the counting quantifier to second-order logic. Let $\mathcal{L}^2_{A,COUNT}$ be $\mathcal{L}^2_A$ together with a new quantifier $\exists s\ x < t$. The intended meaning of $\exists s\ x < t\varphi(x)$ is that there are exactly s values of $x < t$ such that $\varphi(x)$ holds. Formally, let $\varphi(x)$ be a formula which may have other free variables than x, and let t, s be number terms not containing x. Then

$$\exists s\ x < t\varphi(x)$$

is a formula, which is true if and only if there are exactly s values of $x < t$ such that $\varphi(x)$ holds.

Without ambiguity, let $\underline{\mathbb{N}}_2$ be the standard model in $\mathcal{L}^2_{A,COUNT}$. Let the definition of a formula representing a relation as discussed in Section 2.2.2. Definition of Σ_0^B formula extends by regarding the counting quantifier as a number quantifier. In particular, let $\Sigma_0^{B,COUNT}$ formulas be those without string quantifiers, and all number quantifiers are

[2]The Σ_1^B formulas correspond to (in first-order logic) strict Σ_1^b formulas, while $g\Sigma_1^B$ formulas correspond to Σ_1^b formulas. Similar for Π_1^B and $g\Pi_1^B$ formulas. Here, g stands for "general".

bounded. It should be straightforward that a relation is in $\mathbf{TC}^0$ if it is represented by a $\Sigma_0^{B,COUNT}$ formula. The converse is also true.

Theorem 2.9 *(*$\Sigma_0^{B,COUNT}$ **Representation Theorem***) A relation is in* $\mathbf{TC}^0$ *if and only if it is represented by a* $\Sigma_0^{B,COUNT}$ *formula.*

Proof: The proof is similar to that of the Σ_0^B Representation Theorem in [12]. We have to translate between $\mathcal{L}_{\mathcal{FO},COUNT}$ sentences and $\Sigma_0^{B,COUNT}$ formulas. Translating $\Sigma_0^{B,COUNT}$ formulas into $\mathcal{L}_{\mathcal{FO},COUNT}$ sentence can be done as follows. For each tuple $(\overline{x}, \overline{Y})$, imagine encoding it into a string $Z = [\overline{x}, \overline{Y}]$ as discussed in Remark 2.4. Now consider the relation represented by a $\Sigma_0^{B,COUNT}$ formula $\varphi(\overline{x}, \overline{Y})$. Each string variable Y_j is encoded in a part of Z, thus each atomic formula $Y_j(t)$ in φ can be translated using appropriate atomic formulas of the form $Z(t')$. Consequently, the relation can be defined by a $\mathcal{L}_{\mathcal{FO},COUNT}$ sentence.

Conversely, suppose that an $\mathcal{L}_{\mathcal{FO},COUNT}$ sentence θ defines a relation $R(\overline{x}, \overline{Y})$, then we need to construct a $\Sigma_0^{B,COUNT}$ formula $\varphi(\overline{x}, \overline{Y})$ which represents R. Essentially, for each bit $Z(t)$, we need to check whether it belongs to the encoding of a string Y_j or a number x_i. If t belongs to the encoding of a number, then since numbers are coded as unary strings, there is not much to be done. Otherwise, if t belong to the encoding of a string Y_j, then $Z(t)$ can be calculated from $Y_j(t')$, for some appropriate t'. The formal proof is as follows.

Sufficiency: Let R be the relation represented by a $\Sigma_0^{B,COUNT}$ formula $\varphi(\overline{x}, \overline{Y})$. With a pairing function (such as $\langle x, y\rangle = (x+y)(x+y+1)+x$), we can assume that every bounding term in φ is $m = \Sigma|Y| + \Sigma x$. Let $Z = [\overline{x}, \overline{Y}]$. Recall that according to the encoding in Remark 2.4, the length of Z is $2(\sum x_i + \sum |Y|_j + k + l + 1)$, where k, l are the numbers of x_i's and Y_j's respectively. An $\mathcal{L}_{\mathcal{FO},COUNT}$ sentence which defines R is of the form

$$\exists a_1, \ldots, a_k, m_1, \ldots, m_l[max = 2(\sum a_i + \sum m_j + k + l + 1) \wedge Locate(\overline{a}, \overline{m}) \wedge \varphi']$$

where a_i is the intended value of x_i ($1 \leq i \leq k$), and m_j is the intended value of $|Y_j|$ ($1 \leq j \leq l$). The formula $Locate(\overline{a}, \overline{m})$ specifies the exact position of the delimiters \$, % as in Remark 2.4. For example, % is encoded as two bits 11 at position $2s$ and $2s+1$, where $s = \sum a_i + k$. The formula φ' is obtained from φ as follows. First, replace each x_i by a_i, $|Y_j|$ by m_j ($1 \leq i \leq k, 1 \leq j \leq l$). Let t' be the term obtained from t by doing this. Then, each occurrence of $Y_j(t)$ is replaced by the formulas saying that at the corresponding positions in Z, the bits properly encode $Y_j(t)$. In particular, since 1 is encoded as 01, $Y_j(t)$ is replaced by $\neg Z(2(t' + s_j)) \wedge Z(2(t' + s_j) + 1)$, where s_j is the offset of the encoding of Y_j in Z. Last, each quantifier $\exists x < m$ ($\forall x < m$, $\exists s\ x < m$ respectively) in φ is replaced by $\exists x$ ($\forall x$, $\exists s\ x$, respectively).

Necessity: Suppose the underlying language $\{[\overline{x}, \overline{Y}] : R(\overline{x}, \overline{Y})\}$ is defined by an $\mathcal{L}_{\mathcal{FO},COUNT}$ sentence θ. We translate θ into an $\mathcal{L}^2_{A,COUNT}$ formula $\varphi(\overline{x}, \overline{Y})$ so that φ represents R. Our main concern is to translate the atomic formula of the form $Z(t)$. Recall that numbers are coded as unary strings, and after we have concatenated the unary strings and binary strings (together with the delimiters), 0 is replaced by 00, 1 is replaced by 01, etc. The atomic formula $Z(t)$ can be translated into a disjunction of the form

$$\bigvee_{i=0}^{k+l} InDelimiter_i(t) \vee \bigvee_{i=1}^{k} InNumber_i(t) \vee \bigvee_{j=1}^{l} InString_j(t).$$

First, $InDelimiter(t)$ checks if t belongs to the coding of any delimiter (there are $k+l+1$ of them), and outputs the appropriate value. In particular, $InDelimiter_0(t)$ checks if t is in the coding of % (i.e., $2(k+x_1+\ldots+x_k) \leq t < 2(k+x_1+\ldots+x_k)+2$), and if so, outputs TRUE (since % is coded by 11). For $1 \leq i \leq k$, $InDelimiter_i(t)$ checks if t is in the coding of the delimiter \$ for x_i (i.e., $2(i-1+x_1+\ldots+x_{i-1}) \leq t < 2(i-1+x_1+\ldots+x_{i-1})+2$), and if so, outputs TRUE if t is even (since \$ is coded by 10). For $k+1 \leq i \leq k+l$, $InDelimiter_i(t)$ is defined similarly.

Next, $InNumber_i(t)$ checks if t belongs to the part of the string Z that codes x_i, and it is true only if t is odd (since 1 is coded by 01). In particular,

$$InNumber_i(t) \equiv (2(i + x_1 + \ldots + x_{i-1}) \leq t < 2(i + x_1 + \ldots + x_i)) \wedge \exists z(2z + 1 = t).$$

Similarly, $InString_j(t)$ checks if t is in the part of Z that codes Y_j, and if so, returns $Y_j(t')$ if t is odd (recall that 1 is coded by 01), for appropriate t'. Formally,

$$InString_j(t) \equiv 2s \leq t < 2(s + |Y_j|) \wedge \exists z(t = 2z + 1 \wedge Y_j(z - s)),$$

where $2s$ is the offset of the encoding of Y_j in Z, i.e., $s = k + j + \sum x_i + |Y_1| + \ldots + |Y_{j-1}|$.

It remains to map the quantifiers in θ to the quantifiers in φ. This should be obvious. The bounds for quantifiers in φ is $2(1 + k + l + \sum x_i + \sum |Y_j|)$. ■

From Theorem 2.9, we are able to derive the closure of $\mathbf{TC}^0$ relations under operations which correspond to the Boolean connectives and the quantifiers (some of these may subsume others). First, we define the operation corresponding to the counting quantifier.

Definition 2.10 *(***The Counting Operation***) Let $Q(i)$ be a relation which might contain other parameters. The relation obtained from $Q(i)$ by applying the counting operation on i is*

$$(\#iQ)(k, j) \text{ iff } k = |\{i : i < j \text{ and } Q(i)\}|.$$

Now, it is straightforward from Theorem 2.9 that $\mathbf{TC}^0$ is closed under Boolean, bounded quantification and counting operations. (Bounded quantification operations can be seen as a special case of the counting operation.) Note that the relations represented by Σ_0^B formulas are exactly $\mathbf{AC}^0$ relations.

Theorem 2.11 *(***Structure of $\mathbf{TC}^0$***) The class of $\mathbf{TC}^0$ relations is the closure of $\mathbf{AC}^0$ relations under Boolean and counting operation.* ■

Corollary 2.12 *The class of $\mathbf{TC}^0$ relations is closed under bounded quantifications* ■

We can have a simpler characterization of the $\mathbf{TC}^0$ relations. This will simplify our proofs in later parts, especially when they are by induction on the structure of $\mathbf{TC}^0$ relations. Let *atomic* relations be the relations of the form

$$p(\overline{x}, |\overline{Y}|) = q(\overline{x}, |\overline{Y}|),\ p(\overline{x}, |\overline{Y}|) < q(\overline{x}, |\overline{Y}|) \text{ and } X(i),$$

for polynomials p, q with natural number coefficients. Then it is straightforward from the $\Sigma_0^{B,COUNT}$ Representation Theorem (Theorem 2.9) that the class of $\mathbf{TC}^0$ relations is the closure of these atomic relations under Boolean and counting operations.

Corollary 2.13 *The class of* $\mathbf{TC}^0$ *relations is the closure of atomic relations under Boolean and counting operations.* ■

2.3 The Class $\mathbf{FTC}^0$ of $\mathbf{TC}^0$ Functions

Intuitively, $\mathbf{FTC}^0$ is the class of number and string functions that can be computed by an **FO**-uniform family of threshold circuits. Here, we define $\mathbf{FTC}^0$ in terms of the class $\mathbf{TC}^0$. The string functions in $\mathbf{FTC}^0$ are polynomially bounded, with bitgraph in $\mathbf{TC}^0$, and the number functions in $\mathbf{FTC}^0$ are polynomially bounded, with graph in $\mathbf{TC}^0$. Here, the arguments of the functions are encoded by the same method as discussed in Remark 2.4. Formal definition is given below.

Definition 2.14 *(***The Class** $\mathbf{FTC}^0$*) A string functions* $F(\overline{x}, \overline{Y})$ *is in* $\mathbf{FTC}^0$ *if for some polynomial* p *and* $\mathbf{TC}^0$ *relation* R,

$$F(\overline{x}, \overline{Y})(i) \Leftrightarrow i < p(\overline{x}, |\overline{Y}|) \wedge R(i, \overline{x}, \overline{Y}),$$

A number function $f(\overline{x}, |\overline{Y}|)$ *is in* $\mathbf{FTC}^0$ *if for some polynomial* p *and* $\mathbf{TC}^0$ *relation* R,

$$f(\overline{x}, |\overline{Y}|) = \mathtt{min}\ z < p(\overline{x}, |\overline{Y}|) R(\overline{x}, \overline{Y}, z),$$

i.e., it is the least number $z < p(\overline{x}, |\overline{Y}|)$ *satisfying* $R(\overline{x}, \overline{Y}, z)$*, or 0 if there is no such* z.

We will now prove some basic properties of $\mathbf{TC}^0$ functions. In particular, they can be substituted for variables in $\mathbf{TC}^0$ relations, and they are closed under composition.

Theorem 2.15 *The class of* $\mathbf{TC}^0$ *relations is closed under substitution of* $\mathbf{TC}^0$ *functions for variables.*

Proof: Let $f(\overline{x}, |\overline{Y}|)$ be a $\mathbf{TC}^0$ number function, then its graph, $z = f(\overline{x}, |\overline{Y}|)$, is in $\mathbf{TC}^0$. Consequently, substituting a $\mathbf{TC}^0$ number function for a number variable in a $\mathbf{TC}^0$ relation clearly results in a $\mathbf{TC}^0$ relation.

Next, consider the case of substitution of string functions for string variables. Corollary 2.13 provides a recursive characteristics of $\mathbf{TC}^0$ relations. We will prove by induction on the formation of a $\mathbf{TC}^0$ relation $Q(\overline{x}, \overline{X}, Z)$ that the relation $R(\overline{x}, \overline{y}, \overline{X}, \overline{Y})$ is also in $\mathbf{TC}^0$, where

$$R(\overline{x}, \overline{y}, \overline{X}, \overline{Y}) \Leftrightarrow Q(\overline{x}, \overline{X}, F(\overline{x}, \overline{y}, \overline{X}, \overline{Y})),$$

for string function F in $\mathbf{FTC}^0$. For readability, we will ignore the variables $\overline{x}, \overline{X}$.

The base case (Q is an atomic relation) is straightforward. Consider, for example,

$$Q(Z) \Leftrightarrow p(|Z|) = q(|Z|), \quad \text{and} \quad F(\overline{y}, \overline{Y})(l) \leftrightarrow l < r(\overline{y}, |\overline{Y}|) \wedge S(l, \overline{y}, \overline{Y}),$$

for some polynomials p, q, r and $\mathbf{TC}^0$ relation S. Then it is easy to obtain $R(\overline{y}, \overline{Y})$ (i.e., $Q(F(\overline{y}, \overline{Y}))$) from Q, S and some atomic relations (we need to differentiate two cases: (i) when $|F(\overline{y}, \overline{Y})| = 0$, and (ii) when $|F(\overline{y}, \overline{Y})| > 0$).

For the induction step, consider the non-trivial case of the counting operation. Suppose $Q(i, Z)$ is a $\mathbf{TC}^0$ relation. Consider the relation obtained from Q by applying the counting operation:

$$(\#iQ)(k, j, Z) \Leftrightarrow k = |\{i < j : Q(i, Z)\}|.$$

Suppose $F(k, j, \overline{y}, \overline{Y})$ is in $\mathbf{FTC}^0$. We need to show that R is in $\mathbf{TC}^0$, where

$$\begin{aligned} R(k, j, \overline{y}, \overline{Y}) &\Leftrightarrow (\#iQ)(k, j, \overline{y}, \overline{Y}, F(k, j, \overline{y}, \overline{Y})) \\ &\Leftrightarrow k = |\{i < j : Q(i, F(k, j, \overline{y}, \overline{Y}))\}| \\ &\Leftrightarrow \exists l \leq j[k = l \wedge l = |\{i < j : Q(i, F(k, j, \overline{y}, \overline{Y}))\}|]. \end{aligned}$$

By the induction hypothesis, the relation $l = |\{i < j : Q(i, F(k, j, \overline{y}, \overline{Y}))\}|$ is in $\mathbf{TC}^0$. Hence $R(k, j, \overline{y}, \overline{Y})$ is in $\mathbf{TC}^0$. ■

Corollary 2.16 *A number function $f(\overline{x}, |\overline{Y}|)$ is in $\mathbf{FTC}^0$ if and only if it is equal to $|F(\overline{x}, \overline{Y})|$ for some string function F in $\mathbf{FTC}^0$.*

Proof: First, let $f(\overline{x}, \overline{Y})$ be a number function in $\mathbf{FTC}^0$, we construct a string function $F(\overline{x}, \overline{Y})$ in $\mathbf{FTC}^0$ so that $f = |F|$. Essentially, $F(\overline{x}, \overline{Y})$ contains only one element, which is $f(\overline{x}, \overline{Y}) - 1$. Suppose that

$$f(\overline{x}, |\overline{Y}|) = \mathtt{min}\ z < p(\overline{x}, |\overline{Y}|) R(\overline{x}, \overline{Y}, z)$$

for some polynomial p and $\mathbf{TC}^0$ relation R. Define

$$F(\overline{x}, \overline{Y})(i) \leftrightarrow i + 1 < p(\overline{x}, \overline{Y}) \wedge R(\overline{x}, \overline{Y}, i + 1) \wedge \forall z \leq i \neg R(\overline{x}, \overline{Y}, z).$$

By Theorem 2.11, Corollary 2.12 and Theorem 2.15, $F(\overline{x}, \overline{Y})(i)$ is a $\mathbf{TC}^0$ relation. It is clear that $f(\overline{x}, \overline{Y}) = |F(\overline{x}, \overline{Y})|$.

Conversely, let $F(\overline{x}, \overline{Y})$ be a string function in $\mathbf{FTC}^0$. Suppose that

$$F(\overline{x}, \overline{Y})(i) \Leftrightarrow i < p(\overline{x}, \overline{Y}) \wedge R(i, \overline{x}, \overline{Y}),$$

where R is a $\mathbf{TC}^0$ relation. Then

$$|F(\overline{x}, \overline{Y})| = \mathtt{min}\ z \leq p(\overline{x}, \overline{Y}) \forall i < p(\overline{x}, \overline{Y})[R(i, \overline{x}, \overline{Y}) \supset i < z].$$

■

It is also immediate from Theorem 2.15 that the $\mathbf{TC}^0$ functions are closed under composition.

Corollary 2.17 *The class $\mathbf{FTC}^0$ is closed under composition.* ■

Chapter 3

The Theory $\mathbf{VTC}^0$

In this chapter, we will first present the second-order theory $\mathbf{VTC}^0$, then show that it characterizes precisely $\mathbf{TC}^0$, i.e., $\mathbf{FTC}^0$ is exactly the class of Σ_1^B-definable functions of $\mathbf{VTC}^0$, and $\mathbf{TC}^0$ relations are exactly Δ_1^B-definable relations in $\mathbf{VTC}^0$. Lastly, we will give an example of reasoning in $\mathbf{VTC}^0$ by proving the Pigeon Hole Principle (*PHP*) in $\mathbf{VTC}^0$. The theory $\mathbf{VTC}^0$ is $\mathbf{V}^0$ together with the *NUMONES* axiom, which ensures the "counting ability". The theory $\mathbf{V}^0$, though it has been considered in other works [26], is first so named in [12]. It has been discussed in detail in [12]. In particular, it has been shown that $\mathbf{V}^0$ characterizes $\mathbf{AC}^0$. Thus *NUMONES* can be seen as a lift from $\mathbf{AC}^0$ to $\mathbf{TC}^0$.

Showing that $\mathbf{VTC}^0$ characterizes $\mathbf{FTC}^0$ involves showing that the Σ_1^B-definable functions of $\mathbf{VTC}^0$ are in $\mathbf{FTC}^0$, and conversely, each Σ_1^1 theorem (recall the definition of Σ_1^1 formulas in Notation 2.8) of $\mathbf{VTC}^0$ can be witnessed by a function in $\mathbf{FTC}^0$. Similar characterizations of complexity classes by logical theories often use the recursive characterization of the function class (the so-called "function algebras"). In particular, the first-order theories $\mathbf{TTC}^0$ [11], $\overline{\mathbf{R}}^0$ [19] and Δ_1^b-$\mathbf{CR}$ [20] have been shown to characterize $\mathbf{TC}^0$ using the fact that $\mathbf{FTC}^0$ (as a class of functions on numbers) is the closure of some initial functions (including multiplication) under composition and CRN (concatenation recursion on notation). Here, we use the recursive characterization of the $\mathbf{TC}^0$ *relations*. One reason is that second-order theories often do not contain multiplication for strings in the first place. Another reason is that, since $\mathbf{FTC}^0$ is defined in terms of $\mathbf{TC}^0$, proving properties of $\mathbf{TC}^0$ functions eventually boils down to proving properties of $\mathbf{TC}^0$ relations. In short, to prove the first direction (i.e., that the $\mathbf{TC}^0$ functions are Σ_1^B-definable in $\mathbf{VTC}^0$), we show that the $\mathbf{TC}^0$ relations are strongly Δ_1^B-definable in $\mathbf{VTC}^0$. Then we prove a general relationship between functions and their bitgraphs: a function is Σ_1^B-definable in a bounded theory if and only if it is bounded, and its bitgraph is strongly Δ_1^B-definable in the theory.

The second direction (i.e., proving the witnessing theorem for $\mathbf{VTC}^0$) requires, first of all, the defining axioms for functions in $\mathbf{FTC}^0$. These axioms are defined inductively from $\mathbf{AC}^0$ functions and *numones* (which witnesses the *NUMONES* axiom). We will actually prove a more general result: if $\mathcal{T}$ is a Σ_0^B theory over a "rich" vocabulary $\mathcal{L}$, then its $g\Sigma_1^B$ theorems can be witnessed by functions in $\mathcal{L}$.

Organization of the chapter is as follows. First, we recall the theory $\mathbf{V}^0$ and introduce $\mathbf{VTC}^0$. Then we show that the $\mathbf{TC}^0$ functions are Σ_1^B-definable in $\mathbf{VTC}^0$ (Corollary 3.17). Next, we will show that Σ_1^B theorems of $\mathbf{VTC}^0$ are witnessed by $\mathbf{FTC}^0$ functions (Corollary 3.25). At the end of the chapter, we will show how to prove *PHP* in $\mathbf{VTC}^0$.

3.1 The Theory $\mathbf{VTC}^0$

Recall the syntax and semantic of second-order logic presented in Section 2.2.1. The theory $\mathbf{V}^0$ is defined as the universal closure of the following axioms: **B1-B14**, **L1, L2, SE** (**SE** stands for string equality); and the Σ_0^B comprehension axiom scheme (Σ_0^B *COMP*). First, let Φ be a set of formulas, then Φ *COMP* is the set of all formulas of the form

$$\exists X \leq y \forall z < y[X(z) \leftrightarrow \varphi(z)], \tag{3.1}$$

where φ is a formula in Φ, and X does not occur in φ. When Φ is the set of all $\Sigma_0^B(\mathcal{L})$ formulas, we refer to Φ *COMP* as $\Sigma_0^B(\mathcal{L})$ *COMP*. Next, the axioms **B1-B14** and **L1, L2, SE** are as follows.

B1 $x + 1 \neq 0$

B2 $x + 1 = y + 1 \supset x = y$

B3 $x + 0 = x$

B4 $x + (y + 1) = (x + y) + 1$

B5 $x \cdot 0 = 0$

B6 $x \cdot (y + 1) = (x \cdot y) + x$

B7 $x \leq x + y$

B8 $(x \leq y \wedge y \leq x) \supset x = y$

B9 $0 + 1 = 1$

B10 $0 \leq x$

B11 $x \leq y \wedge y \leq z \supset x \leq z$

B12 $x \leq y \vee y \leq x$

B13 $x \leq y \leftrightarrow x < y + 1$

B14 $x \neq 0 \supset \exists y(y + 1 = x)$

L1 $X(y) \supset y < |X|$

L2 $y + 1 = |X| \supset X(y)$

SE $X = Y \leftrightarrow [|X| = |Y| \wedge \forall i < |X|(X(i) \leftrightarrow Y(i))]$

Remark 3.1 *From now on, when we mention a (second-order) theory in general, we tacitly assume that it contains the axioms* **B1–B14***,* **L1, L2** *and* **SE***.*

Definition 3.2 *(***The Theory $\mathbf{V}^0$ [12]***) The theory $\mathbf{V}^0$ is the universal closure of the axioms* **B1-B14, L1, L2, SE***, together with the Σ_0^B COMP axiom scheme.*

The theory $\mathbf{V}^0$ has been shown to characterize $\mathbf{AC}^0$. It has also been shown [12] that $\mathbf{V}^0$ is conservative over $I\Delta_0$. As a result, the functions which are definable in $I\Delta_0$ can be used in $\mathbf{V}^0$. Furthermore, they can appear in the comprehension axiom scheme, i.e., if we extend $\mathcal{L}_A^2$ to the vocabulary $\mathcal{L}$ which contains these symbols, and extend $\mathbf{V}^0$ to contain their defining axioms, as well as replacing the Σ_0^B *COMP* axiom scheme by the $\Sigma_0^B(\mathcal{L})$ *COMP* induction scheme, then we obtain a conservative extension of $\mathbf{V}^0$.[1] In

[1]The issue of introducing new symbols will be dealt with later, in Section 3.3.1

particular, we will assume that the pairing function $\langle x, y\rangle$ is in the language, where

$$2\langle x, y\rangle = (x+y)(x+y+1) + 2x.$$

The important property of this pairing function is that it is a injection from $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$, and its inverse is easily computed. We will denote $X(\langle x, y\rangle)$ simply by $X(x, y)$. This provides a mechanism for coding multiple strings into an "array". For example, suppose that Y is the intended array, then its ith row can be retrieved by

$$Y^{[i]}(j) \leftrightarrow Y(i, j).$$

In addition, let Φ be a set of formula, then the number induction scheme for Φ is the set of formula of the form

$$[\varphi(0) \wedge \forall x < z(\varphi(x) \supset \varphi(x+1))] \supset \varphi(z),$$

for $\varphi \in \Phi$. In particular, the $\Sigma_0^B(\mathcal{L})$ ($\Sigma_1^B(\mathcal{L})$) number induction scheme is the above scheme when Φ is the set of all $\Sigma_0^B(\mathcal{L})$ ($\Sigma_1^B(\mathcal{L})$) formulas. It has been shown [12] that Σ_0^B number induction is provable in $\mathbf{V}^0$.

The lift from $\mathbf{AC}^0$ to $\mathbf{TC}^0$ is powered by the axiom *NUMONES*. Essentially, this axiom states that for each string X, there is a "counting array" Y whose ith row contains an unique number, which is the number of bits in X up to (but not including) position i. Formally, let $\varphi_N(X, Y)$ be the Σ_0^B formula expressing that Y is the counting array for X:

$$\varphi_N(X,Y) \equiv \forall i \leq |X| \exists! j \leq |X| Y(i,j)) \wedge Y(0,0) \wedge \forall i < |X| \forall j \leq |X|$$
$$[(Y(i,j) \wedge X(i) \supset Y(i+1, j+1)) \wedge (Y(i,j) \wedge \neg X(i) \supset Y(i+1, j))].$$

Then, *NUMONES* states that for each string X, there is a string Y such that $\varphi_N(X, Y)$ holds.

Definition 3.3 *(***The Axiom** *NUMONES) The axiom NUMONES is defined as* $\forall X \exists Y \leq 1 + \langle |X|, |X|\rangle \varphi_N(X, Y)$.

Definition 3.4 (**The Theory $\mathbf{VTC}^0$**) *The theory* $\mathbf{VTC}^0$ *is* $\mathbf{V}^0$ *together with NUMONES.*

It has been shown [13] that $\mathbf{V}^0$ is finitely axiomatizable. Therefore, $\mathbf{VTC}^0$ is also finitely axiomatizable.

Corollary 3.5 *The theory* $\mathbf{VTC}^0$ *is finitely axiomatizable.* ■

3.2 Definability of $\mathbf{TC}^0$ Functions and Predicates

In this section, we will show that the $\mathbf{TC}^0$ functions are Σ_1^B-definable in $\mathbf{VTC}^0$. First, we recall the definition of Σ_1^B-definable functions. Then we introduce the notion of "strong Δ_1^B-definability" of relations in a theory, and show a relationship between Σ_1^B-definable functions and strongly Δ_1^B-definable predicates of a bounded theory: the Σ_1^B-definable functions of a bounded theory are exactly those whose bitgraphs are bounded and strongly Δ_1^B-definable in the theory. Then we show that the $\mathbf{TC}^0$ relations are strongly Δ_1^B-definable in $\mathbf{VTC}^0$. As a corollary, the $\mathbf{TC}^0$ functions are Σ_1^B-definable in $\mathbf{VTC}^0$.

Definition 3.6 (**Σ_1^B-Definable Functions**) *A string function $F(\overline{x}, \overline{Y})$ is Σ_1^B-definable in a theory $\mathcal{T}$ if there is a Σ_1^B formula $\varphi(\overline{x}, \overline{Y}, Z)$ such that $F(\overline{x}, \overline{Y}) = Z \Leftrightarrow \varphi(\overline{x}, \overline{Y}, Z)$, and that $\mathcal{T} \vdash \forall\overline{x}\forall\overline{Y}\exists!Z\varphi(\overline{x}, \overline{Y}, Z)$.*

A number function $f(\overline{x}, \overline{Y})$ is Σ_1^B-definable in a theory $\mathcal{T}$ if there is a Σ_1^B formula $\varphi(\overline{x}, \overline{Y}, z)$ such that $f(\overline{x}, \overline{Y}) = z \Leftrightarrow \varphi(\overline{x}, \overline{Y}, z)$, and that $\mathcal{T} \vdash \forall\overline{x}\forall\overline{Y}\exists!z\varphi(\overline{x}, \overline{Y}, z)$.

Definition 3.7 (**Σ_0^B-Bit Definable Functions**) *A string function $F(\overline{x}, \overline{Y})$ is Σ_0^B-bit definable in a theory $\mathcal{T}$ if there is a Σ_0^B formula of $\mathcal{T}$ $\varphi(\overline{x}, \overline{Y}, Z)$ such that $F(\overline{x}, \overline{Y})(i) \Leftrightarrow [i < t \wedge \varphi(i, \overline{x}, \overline{Y})]$,*

A number function $f(\overline{x}, \overline{Y})$ is Σ_0^B-definable in a theory $\mathcal{T}$ if there is a string function F which is Σ_0^B-bit definable in $\mathcal{T}$ such that $\mathcal{T} \vdash \forall\overline{x}\forall\overline{Y} f(\overline{x}, \overline{Y}) = |F(\overline{x}, \overline{Y})|$.

Definition 3.8 *(*Δ_1^B**-Definable Relations***) A relation R is Δ_1^B-definable in a theory $\mathcal{T}$ if there is a Σ_1^B formula ϕ_1 and a Π_1^B formula ϕ_2 such that they both represent R, and that $\mathcal{T} \vdash \phi_1 \leftrightarrow \phi_2$.*

We will now define the notion of strong Δ_1^B-definability. Informally, a relation R is strongly Δ_1^B-definable in a theory $\mathcal{T}$ if R is Δ_1^B-definable in $\mathcal{T}$, and furthermore $\mathcal{T}$ can prove the existence of witnesses for membership in R. Notice that using the pairing function $\langle x, y\rangle$, each Σ_1^B (Π_1^B) formula is equivalent to a single-Σ_1^B (Π_1^B) formula.

Definition 3.9 *(***Strongly** Δ_1^B**-Definable Relations***) A relation $R(\overline{x}, \overline{Y})$ is strongly Δ_1^B-definable in a theory $\mathcal{T}$ if there is a Σ_1^B formula $\exists Z \leq t\varphi(\overline{x}, \overline{Y}, Z)$ and a Π_1^B formula $\forall Z \leq t\theta(\overline{x}, \overline{Y}, Z)$ (φ and θ are Σ_0^B formulas), such that they both represent R, and that*

$$\mathcal{T} \vdash \exists Z \leq t\varphi(\overline{x}, \overline{Y}, Z) \leftrightarrow \forall Z \leq t\theta(\overline{x}, \overline{Y}, Z),$$

$$\mathcal{T} \vdash \exists W \leq \langle \overline{b}, t\rangle \forall \overline{x} < \overline{b}\ [|W^{[\overline{x}]}| \leq t \wedge (\varphi(\overline{x}, \overline{Y}, W^{[\overline{x}]}) \vee \neg\theta(\overline{x}, \overline{Y}, W^{[\overline{x}]}))].$$

Note that without loss of generality, we can assume that the Σ_1^B (Π_1^B) formulas in the definitions are single-Σ_1^B (single-Π_1^B). It is clear that if $\mathcal{T}$ admits the Σ_0^B replacement rule (Definition 4.10), then R is strongly Δ_1^B-definable in $\mathcal{T}$ if and only if it is Δ_1^B-definable in $\mathcal{T}$. It follows from Lemma 4.11 that $\mathbf{VTC}^0$ admits Σ_0^B replacement rule, and thus the two notions coincide for $\mathbf{VTC}^0$.

3.2.1 Bounded Theories and Parikh's Theorem

Parikh's Theorem has been generalized for second-order logic in [12]. Here, we need a further generalization, where we consider a theory over vocabularies which may contain string function symbols. It states that if a *bounded theory* $\mathcal{T}$ proves the existence of z in $\exists z\varphi(z, \overline{x}, \overline{Y})$, for a bounded formula φ, then $\mathcal{T}$ also proves that z can be bounded, i.e., $\mathcal{T}$ also proves $\exists z < t\varphi(z, \overline{x}, \overline{Y})$, for some number term t. The proof is identical to the proof presented in [12], with the exception that the rules for introducing second-order quantifications (page 64 [12]) are extended to include string terms. First, we define the concept of bounded theory. It is based on a property called *monotone bounding property* [12], which is satisfied by all of the theories that we will consider.

Definition 3.10 *(***Monotone Bounding Property [12]***) A theory $\mathcal{T}$ in a vocabulary $\mathcal{L}$ has monotone bounding property if for all number terms $r(\overline{a}, \overline{\gamma})$ and $s(b, \overline{a}, \overline{\gamma})$, there is a number term $t(\overline{a}, \overline{\gamma})$ such that $\mathcal{T} \vdash b < r(\overline{a}, \overline{\gamma}) \supset s(b, \overline{a}, \overline{\gamma}) < t(\overline{a}, \overline{\gamma})$.*

Definition 3.11 *(***Bounded Theories***) A formula is bounded if all of its quantifiers (both number and string quantifiers) are bounded. A theory $\mathcal{T}$ is called a bounded theory if it has the monotone bounding property, and it can be axiomatized by bounded axioms.*

Theorem 3.12 *(***Second-order Parikh Theorem [12]***) Suppose that $\mathcal{T}$ is a bounded theory, and φ is a bounded formula such that $\mathcal{T} \vdash \exists z \varphi(\overline{x}, \overline{Y}, z)$. Then there is some number term t such that $\mathcal{T} \vdash \exists z < t(\overline{x}, \overline{Y}) \varphi(\overline{x}, \overline{Y}, z)$* ■

The above theorem states a bound on the number variable z. We can also bound the string variable as follows

Corollary 3.13 *Suppose that $\mathcal{T}$ is a bounded theory, and φ is a bounded formula such that $\mathcal{T} \vdash \exists Z \varphi(\overline{x}, \overline{Y}, Z)$. Then there is some number term t such that $\mathcal{T} \vdash \exists Z < t(\overline{x}, \overline{Y}) \varphi(\overline{x}, \overline{Y}, Z)$*

Proof: Let $\theta(\overline{x}, z, \overline{Y}) \equiv \exists Z \le z \varphi(\overline{x}, \overline{Y}, Z)$. Then θ is a bounded formula, and $\mathcal{T} \vdash \exists z \theta(\overline{x}, z, \overline{Y})$. By Parikh's Theorem, there is a number term $t(\overline{x}, \overline{Y})$ such that $\mathcal{T} \vdash \exists z < t \theta(\overline{x}, z, \overline{Y})$. Thus $\mathcal{T} \vdash \exists Z < t \varphi(\overline{x}, \overline{Y}, Z)$. ■

As a corollary to Corollary 3.13, if the string function $F(\overline{x}, \overline{Y})$ is Σ_1^B-definable in a bounded theory $\mathcal{T}$, then the number function $|F(\overline{x}, \overline{Y})|$ is also Σ_1^B-definable in $\mathcal{T}$. The defining axiom for $|F|$ is the prenex form of $\exists Z \le t \varphi(\overline{x}, \overline{Y}, Z) \wedge z = |Z|$, where φ is the defining axiom for F, and t is the bound on Z, as stated by Corollary 3.13.

Corollary 3.14 *Suppose that $\mathcal{T}$ is a bounded theory, and that $F(\overline{x}, \overline{Y})$ is a Σ_1^B-definable string function in $\mathcal{T}$. Then the number function $|F(\overline{x}, \overline{Y})|$ is also Σ_1^B-definable in $\mathcal{T}$.* ■

3.2.2 Σ_1^B-Definable Functions and Strongly Δ_1^B-Definable Predicates

We will now prove that a string function is Σ_1^B-definable in a bounded theory $\mathcal{T}$ which contains $\mathbf{V}^0$ if its bitgraph is strongly Δ_1^B-definable in $\mathcal{T}$.

Note that we can have a necessary and sufficient condition for a string function to be Σ_1^B-definable in a bounded theory as follows. In Definition 3.9, membership in $R(\overline{x}, \overline{Y})$ is witnessed uniformly with respect to all number arguments (i.e., $\overline{x}$) of R. We can restrict this definition so that $R(\overline{x}, \overline{Y})$ is strongly Δ_1^B-definable in $\mathcal{T}$ *with respect to* $\overline{y}$, where $\overline{y}$ is a subset of $\overline{x}$, if membership in R is witnessed uniformly with respect to $\overline{y}$. Formally, the second condition of Definition 3.9 is modified as

$$\mathcal{T} \vdash \exists W \leq \langle \overline{b}, t \rangle \forall \overline{y} < \overline{b} \; [|W^{[\overline{y}]}| \leq t \wedge (\varphi(\overline{x}, \overline{Y}, W^{[\overline{y}]}) \vee \neg\theta(\overline{x}, \overline{Y}, W^{[\overline{y}]}))].$$

Then, if a predicate R is strongly Δ_1^B-definable in $\mathcal{T}$ then it is also strongly Δ_1^B-definable in $\mathcal{T}$ with respect to each of its number arguments. Now, a stronger result than Lemma 3.15 (below) is that F is Σ_1^B-definable in a bounded theory $\mathcal{T}$ if and only if its bitgraph $F()(i)$ is strongly Δ_1^B-definable in $\mathcal{T}$ with respect to i. However, we will not need this stronger result in this thesis.

Lemma 3.15 *Let $\mathcal{T}$ be a bounded theory which contains $\mathbf{V}^0$. Then a string function F is Σ_1^B-definable in $\mathcal{T}$ if it is bounded, and its bitgraph is strongly Δ_1^B-definable in $\mathcal{T}$.*

Proof: Suppose that F is bounded, and the bitgraph of F is strongly Δ_1^B-definable in $\mathcal{T}$. W.l.o.g., suppose that φ and θ are Σ_0^B formulas, and r, s are some number terms (r depends on $\overline{x}, \overline{X}$, and s depends on $i, \overline{x}, \overline{X}$) such that

$$F(\overline{x}, \overline{X})(i) \quad \Leftrightarrow \quad i < r \wedge \exists Z < s \varphi(i, \overline{x}, \overline{X}, Z) \quad \Leftrightarrow \quad i < r \wedge \forall Z < s \theta(i, \overline{x}, \overline{X}, Z)$$

and

$$\mathcal{T} \vdash \exists Z < s\varphi(i,\overline{x},\overline{X},Z) \leftrightarrow \forall Z < s\theta(i,\overline{x},\overline{X},Z), \tag{3.2}$$

$$\mathcal{T} \vdash \exists W \leq t(b)\forall i < b[|W^{[i]}| \leq s \wedge (\varphi(i,\overline{x},\overline{X},W^{[i]}) \vee \neg\theta(i,\overline{x},\overline{X},W^{[i]}))]. \tag{3.3}$$

Now $Y = F(\overline{x},\overline{X})$ can be collected using Σ_0^B *COMP*, checking for each i whether $\varphi(i,\overline{x},\overline{X},W^{[i]})$ or $\neg\theta(i,\overline{x},\overline{X},W^{[i]})$ holds. The existence of these witnesses is guaranteed by the existence of W in Equation 3.3. Formally,

$$|Y| \leq r \wedge \ \forall i < r\ [Y(i) \leftrightarrow \varphi(i,\overline{x},\overline{X},W^{[i]})].$$

As a result, $\mathcal{T}$ proves the existence of Y. The uniqueness of Y can be proved by Σ_0^B induction. Let $\psi(\overline{x},\overline{X},Y)$ be

$$\exists W \leq t(r)\forall i < r[|W^{[i]}| \leq s\wedge(\varphi(i,\overline{x},\overline{X},W^{[i]})\vee\neg\theta(i,\overline{x},\overline{X},W^{[i]}))\wedge(Y(i) \leftrightarrow \varphi(i,\overline{x},\overline{X},W^{[i]})].$$

Then $F(\overline{x},\overline{X}) = Y \Leftrightarrow \psi(\overline{x},\overline{X},Y)$. Suppose Y', Y'' are such that $\psi(\overline{x},\overline{X},Y')$ and $\psi(\overline{x},\overline{X},Y'')$ hold. Then it is straightforward using Σ_0^B induction that for $i < r$, $Y'(i) \leftrightarrow Y''(i)$. Thus $Y' = Y''$. ■

3.2.3 Definability of $\mathbf{TC}^0$ Functions and Predicates

Lemma 3.16 *The relations in* $\mathbf{TC}^0$ *are strongly* Δ_1^B*-definable in* $\mathbf{VTC}^0$.

Proof: The proof is by induction on the formation of $\mathbf{TC}^0$ relations. Recall from Theorem 2.11 that $\mathbf{TC}^0$ relations are obtained from $\mathbf{AC}^0$ relations by applications of Boolean and counting operations.

The base case is trivial, since $\mathbf{AC}^0$ relations are represented by Σ_0^B formulas.

For the induction step, we consider the non-trivial case of the counting operation. Suppose that the relation $Q(i)$ (which may have other parameters) is strongly Δ_1^B-definable in $\mathbf{VTC}^0$, i.e., there are Σ_0^B formulas $\varphi(i,Z)$ and $\theta(i,Z)$, and a term t such that both $\exists Z \leq t\varphi(i,Z)$ and $\forall Z \leq t\theta(i,Z)$ represent Q, and

$$\mathbf{VTC}^0 \vdash \exists Z \leq t\varphi(i,Z) \leftrightarrow \forall Z \leq t\theta(i,Z),$$

$$\mathbf{VTC}^0 \vdash \exists W \leq r \forall i < b\ [|W^{[i]}| \leq t \wedge (\varphi(i, W^{[i]}) \vee \neg\theta(i, W^{[i]}))], \tag{3.4}$$

where $r = \langle b, t\rangle$. We will present a Σ_1^B formula $\psi_1(k, j)$ and a Π_1^B formula $\psi_2(i, j)$ which both represent $(\#Q)(k, j)$. Note that $(\#Q)(k, j)$ holds exactly when k is the number of i's which are $< j$ and satisfy Q. Let $\psi(j, W, U, V)$ be the formula stating that W is an array of witnesses for either φ or $\neg\theta$, U contains the "flags" indicating whether φ or $\neg\theta$ is being witnessed, and V counts the number of "flags" in U. Then $(\#Q)(k, j)$ holds if there exist W, U, V such that V counts correctly (i.e., $V(j, k)$ holds), and $(\#Q)(k, j)$ also holds if for all such W, U, V, V must count correctly. More precisely, let $\psi(j, W, U, V)$ be

$$\forall i < j\ [|W^{[i]}| \leq t \wedge (\varphi(i, W^{[i]}) \vee \neg\theta(i, W^{[i]})) \wedge (U(i) \leftrightarrow \varphi(i, W^{[i]}))]\ \wedge\ \varphi_N(U, V).$$

Then it follows that $(\#Q)(k, j)$ is represented by both

$$\psi_1(k, j) \Leftrightarrow \exists V \leq \langle j, j\rangle \exists U \leq j \exists W \leq \langle j, t(j)\rangle\ [\psi(j, W, U, V) \wedge V(j, k)],$$

and

$$\psi_2(k, j) \Leftrightarrow \forall V \leq \langle j, j\rangle \forall U \leq j \forall W \leq \langle j, t(j)\rangle\ [\psi(j, W, U, V) \supset V(j, k)].$$

Next, we show that $\mathbf{VTC}^0$ proves that ψ_1 and ψ_2 are equivalent. The direction $\psi_1(k, j) \rightarrow \psi_2(k, j)$ (uniqueness) can be shown in $\mathbf{VTC}^0$ as follows. Suppose that W', U', V' and W'', U'', V'' satisfy $\psi(j, W, U, V)$. Then by Σ_0^B number induction on i we can show that $U' = U''$, hence $V' = V''$. The other direction (existence) is as follows: The existence of W follows from Equation 3.4, U from Σ_0^B *COMP*, and V from *NUMONES*.

It remains to show that the membership in $(\#Q)$ can be witnessed in $\mathbf{VTC}^0$, i.e.,

$$\mathbf{VTC}^0 \vdash \exists V_1 U_1 W_1 \forall k, j < b[\varphi_1(k, j, W_1^{[k,j]}, U_1^{[k,j]}, V_1^{[k,j]}) \vee \neg\theta_1(k, j, W_1^{[k,j]}, U_1^{[k,j]}, V_1^{[k,j]})],$$

where

$$\varphi_1(k, j, W, U, V) \Leftrightarrow \psi(j, W, U, V) \wedge V(j, k), \quad \theta_1(k, j, W, U, V) \Leftrightarrow \psi(j, W, U, V) \supset V(j, k).$$

Again, existence of W_1 follows from Equation 3.4 and Σ_0^B *COMP*. Existence of U_1 is by Σ_0^B *COMP*, and existence of V_1 follows from *NUMONES* axiom as well as Σ_0^B *COMP*. ■

As a corollary, string functions in $\mathbf{FTC}^0$ are Σ_1^B-definable in $\mathbf{VTC}^0$. By Corollary 2.16 and Corollary 3.14, the number functions in $\mathbf{FTC}^0$ are also Σ_1^B-definable in $\mathbf{VTC}^0$.

Corollary 3.17 *The functions in* $\mathbf{FTC}^0$ *are* Σ_1^B*-definable in* $\mathbf{VTC}^0$*.* ■

3.3 Witnessing in $\mathbf{VTC}^0$

First, we discuss the issue of introducing new function and predicate symbols to a bounded theory. We will then prove a more general form of the witnessing theorem, from which we can derive the witnessing theorem for $\mathbf{VTC}^0$.

3.3.1 Introducing New Function and Predicate Symbols

Introducing symbols for definable functions and predicates has been discussed thoroughly in the literature, e.g., [8]. In the context of first-order logic, suppose that a theory $\mathcal{T}'$ over $\mathcal{L}'$ is obtained from a theory $\mathcal{T}$ over $\mathcal{L}$ by adding new symbols (together with their defining axioms) for Σ_1^b-definable functions or Δ_1^b-definable predicates. Then for $i \geq 1$, in $\mathcal{T}'$, each $\Sigma_i^b(\mathcal{L}')$ ($\Pi_i^b(\mathcal{L}')$) formula is equivalent to a $\Sigma_i^b(\mathcal{L})$ ($\Pi_i^b(\mathcal{L})$) formula. Note that Σ_i^b (Π_i^b) formulas in [8] correspond to $g\Sigma_1^B$ ($g\Pi_1^B$) formulas (Notation 2.8).

In our case, we will consider general case where the theory may not contain Σ_0^B replacement axiom scheme. Also, in the context of second-order logic, we need to consider both string functions and number functions. Our goal is to show the following claim. Let $\mathcal{T}$ be a bounded theory which contains some axiom scheme such as $\Sigma_0^B(\mathcal{L})$ *COMP*. Let $\mathcal{L}'$ be the vocabulary $\mathcal{L}$ of $\mathcal{T}$ together with new symbols for some Σ_0^B-bit definable functions and/or strongly Δ_1^B-definable predicates in $\mathcal{T}$. Let $\mathcal{T}'$ be $\mathcal{T}$ together with the defining axioms of the new symbols, and $\Sigma_0^B(\mathcal{L}')$ *COMP* instead of $\Sigma_0^B(\mathcal{L})$ *COMP*. Then $\mathcal{T}'$ is conservative over $\mathcal{T}$. This claim follows immediately from the following theorem.

Theorem 3.18 (**Translation Theorem**) *Suppose $\mathcal{T}$ is a bounded theory over $\mathcal{L}$, and let $\mathcal{T}'$ result from $\mathcal{T}$ by adding a $\Sigma_0^B(\mathcal{L})$-bit definable function or a strongly Δ_1^B-definable predicate (with defining axioms), and let $\mathcal{L}'$ be the expanded language. Then any $\Sigma_0^B(\mathcal{L}')$ formula provably (in $\mathcal{T}'$) defines a strongly Δ_1^B-definable predicate in $\mathcal{T}$.*

Note that by this theorem, each $\Sigma_1^B(\mathcal{L}')$ (respectively $\Pi_1^B(\mathcal{L}')$) formula is equivalent in $\mathcal{T}'$ to a $\Sigma_1^B(\mathcal{L})$ (respectively $\Pi_1^B(\mathcal{L})$) formula.

Proof: Then the proof is straightforward by induction on the structure of the defining axioms. We may need to use Parikh's Theorem to get a bound on the values of the definable functions. ■

3.3.2 Witnessing Theorems

Our goal is to prove a witnessing theorem for $\mathbf{VTC}^0$: for every theorem $\exists Z\varphi(\overline{x}, \overline{Y}, Z)$ of $\mathbf{VTC}^0$, where φ is a Σ_0^B formula, there is a $\mathbf{TC}^0$ function F such that $\varphi(\overline{x}, \overline{Y}, F(\overline{x}, \overline{Y}))$ holds in the conservative extension of $\mathbf{VTC}^0$, which is obtained from $\mathbf{VTC}^0$ by adding the proper functions symbols and their defining axioms. We will prove a more general theorem, which entails the desired result for $\mathbf{VTC}^0$.

In order to state a witnessing theorem for a theory $\mathcal{T}$, i.e., that some class of theorems of $\mathcal{T}$ can be witnessed by functions in a class $\mathcal{C}$, clearly $\mathcal{T}$ must be be able to define functions in $\mathcal{C}$. Here, we will start with the assumption that the vocabulary of $\mathcal{T}$ already contains symbols for all functions in $\mathcal{C}$, and that $\mathcal{T}$ already contains their defining axioms. We make this precise in the concept of "a vocabulary is closed under bitgraph expansion in a theory" below.

Definition 3.19 *Suppose that $\mathcal{L}$ is a vocabulary extending $\mathcal{L}_A^2$, and $\mathcal{T}$ is a theory over $\mathcal{L}$. Then $\mathcal{L}$ is said to be closed under bitgraph expansion in $\mathcal{T}$ if (i) for any $\mathcal{L}$ number* *term* $t(\overline{x}, \overline{Y})$, *and any* $\Sigma_0^B(\mathcal{L})$ *formula* $\varphi(i, \overline{x}, \overline{Y})$, *there is a string function symbol* $F_{t,\varphi}$ *in* $\mathcal{L}$ *such that* $\mathcal{T} \vdash \forall i\ F_{t,\varphi}(\overline{x}, \overline{Y})(i) \leftrightarrow [i < t(\overline{x}\overline{Y}) \wedge \varphi(i, \overline{x}, \overline{Y})]$; *and (ii) for each string function symbol* F *in* $\mathcal{L}$, *there is a number function symbol* f *in* $\mathcal{L}$ *such that* $\mathcal{T} \vdash f = |F|$.

A theory is Σ_0^B if it is axiomatizable by Σ_0^B axioms. First, we will prove that the existence of Y in any theorem of the form $\exists Y \varphi(Y)$ (where φ is Σ_0^B) of a Σ_0^B theory $\mathcal{T}$ can be witnessed by a function of $\mathcal{T}$. From this result, we can get the witnessing theorem for $\mathbf{VTC}^0$ (Corollary 3.25). Then, we will show the same result, but in the case where φ is a $g\Sigma_1^B$ formula.

Theorem 3.20 *(*Σ_0^B **Witnessing Theorem***) Suppose that* $\mathcal{T}$ *is a* Σ_0^B *theory over the vocabulary* $\mathcal{L}$ *which is closed under bitgraph expansion in* $\mathcal{T}$. *Then for each* Σ_1^1 *theorem* $\forall\overline{x}\forall\overline{Y}\exists Z\varphi(\overline{x}, \overline{Y}, Z)$ *of* $\mathcal{T}$ *(where* φ *is* $\Sigma_0^B(\mathcal{L})$*), there is a string function symbol* F *in* $\mathcal{L}$ *such that* $\mathcal{T} \vdash \forall\overline{x}\forall\overline{Y}\varphi(\overline{x}, \overline{Y}, F(\overline{x}, \overline{Y}))$.

Proof: The proof is similar to the proof of the Witnessing Theorem for $\mathbf{V}^0$ [12], and is by induction on the length of the anchored $\mathbf{LK}^2$-$\mathcal{T}$ proof (i.e., the $\mathbf{LK}^2$ proof where axioms of $\mathcal{T}$ can be used) of the theorem of $\mathcal{T}$. For simplicity, we will concern only with single-Σ_1^1 theorem of $\mathcal{T}$. First, note that by the hypothesis, $\mathcal{L}$ is closed under bitgraph expansion in $\mathcal{T}$, thus each string $\mathcal{L}$ term T actually denotes a string function in $\mathcal{L}$.

Remark 3.21 *For each string* $\mathcal{L}$ *term* T, *there is a string function symbol* F *in* $\mathcal{L}$ *such that* $\mathcal{T} \vdash F = T$.

Suppose that $\exists Z\varphi(\overline{x}, \overline{Y}, Z)$ is a theorem of $\mathcal{T}$, where φ is Σ_0^B. Let π be an anchored $\mathbf{LK}^2$-$\mathcal{T}$ proof of $\rightarrow \exists Z\varphi(\overline{x}, \overline{Y}, Z)$, then every formula in π is either Σ_0^B or single-Σ_1^1. Thus sequents in π have the form

$$\mathcal{S} = \quad \exists X_1\phi_1(X_1), \ldots, \exists X_m\phi_m(X_m), \Lambda \longrightarrow \Gamma, \exists Y_1\psi_1(Y_1), \ldots, \exists Y_n\psi_n(Y_n), \tag{3.5}$$

for $m, n \geq 0$; and ϕ_i, ψ_j and the formulas in Γ, Δ are Σ_0^B. We will prove by induction on the depth of $\mathcal{S}$ in π that there are function symbols $F_1, \ldots, F_n$ in $\mathcal{L}$ such that $\mathcal{T}$ proves $\mathcal{S}'$, where

$$\mathcal{S}' = \quad \phi_1(\beta_1), \ldots, \phi_m(\beta_m), \Lambda \longrightarrow \Gamma, \psi_1(F_1(\overline{a}, \overline{\alpha}, \overline{\beta})), \ldots, \psi_n(F_n(\overline{a}, \overline{\alpha}, \overline{\beta})), \tag{3.6}$$

where $\overline{a}, \overline{\alpha}$ are the free variables appearing in $\mathcal{S}$. The case where $\mathcal{S}$ is an axiom of $\mathcal{T}$ is trivial. Other cases are as follows.

Case I (**String $\exists$ right**): Suppose that $\mathcal{S}$ is the bottom sequent of the inference

$$\frac{\Lambda \longrightarrow \Gamma, \varphi(T)}{\Lambda \longrightarrow \Gamma, \exists Y \varphi(Y)},$$

where φ is a Σ_0^B formula. By Remark 3.21, there is a string function F in $\mathcal{L}$ such that $\mathcal{T} \vdash F = T$. Therefore, the witnessing function for Y can be taken as F. Note that if some free variables are eliminated due to the replacement of T, we can substitute the constants 0 or $\emptyset$ for their values in the functions F_j's.

Case II (**String $\exists$ left**): No new function is needed.

Case III (**Number $\exists$ right and $\forall$ left**): No new function is needed. Again, if some free variables are eliminated due to the replacement of T, we can substitute the constants 0 or $\emptyset$ for their values in the functions F_j's.

Case IV (**Number $\exists$ left and number $\forall$ right**): We consider number $\exists$ left, since number $\forall$ right is similar. Suppose that $\mathcal{S}$ is the bottom sequent of the inference

$$\frac{\mathcal{S}_1}{\mathcal{S}} = \frac{b < t \wedge \varphi(b), \Lambda \longrightarrow \Gamma}{\exists x < t\varphi(x), \Lambda \longrightarrow \Gamma},$$

where φ is a Σ_0^B formula, and b does not appear in $\mathcal{S}$. The witnessing functions for $\mathcal{S}$ are obtained from that of $\mathcal{S}_1$ by replacing the free variable b by the following function $g(\overline{a}, \overline{\alpha}) = min\ b < t\varphi(b)$. Note that there is a string function G in $\mathcal{L}$ such that $\mathcal{T} \vdash \forall i\ G(\overline{a}, \overline{\alpha})(i) \leftrightarrow [i < t \wedge \forall j < i \neg\varphi(i)]$. Then $g = |G|$ is a symbol in $\mathcal{L}$. By Remark 3.21 and the induction hypothesis, there are function symbols in $\mathcal{L}$ that witness $\mathcal{S}$.

Case Va (**Cut** Σ_0^B **formulas**): Suppose that $\mathcal{S}$ is the bottom sequent of the inference

$$\frac{\mathcal{S}_1 \quad \mathcal{S}_2}{\mathcal{S}} = \frac{\Lambda \longrightarrow \Gamma, \varphi \qquad \Lambda, \varphi \longrightarrow \Gamma}{\Lambda \longrightarrow \Gamma},$$

where φ is a Σ_0^B formula. Let F_j^1's and F_j^2's be the witnessing functions for Γ in $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. Then the witnessing function F_j for $\mathcal{S}$ can be taken as

$$F_j()(i) \leftrightarrow [(\neg\varphi \wedge F_j^1()(i)) \vee (\varphi \wedge F_j^2()(i))].$$

Note that F_j's are in $\mathcal{L}$, and $\mathcal{T}$ contains their defining axioms.

Case Vb (**Cut** Σ_1^1 **formulas**): Suppose that $\mathcal{S}$ is the bottom sequent of the inference

$$\frac{\mathcal{S}_1 \quad \mathcal{S}_2}{\mathcal{S}} = \frac{\Lambda \longrightarrow \Gamma, \exists Y \varphi(Y) \qquad \Lambda, \exists Y \varphi(Y) \longrightarrow \Gamma}{\Lambda \longrightarrow \Gamma},$$

where φ is a Σ_0^B formula. By the induction hypothesis, there is a witnessing function G for φ in $\mathcal{S}_1$. Also, the witnessing functions in $\mathcal{S}_2$ has one more argument than those in $\mathcal{S}_1$ and $\mathcal{S}$. Let F_j^1 and $F_j^2(\beta)$ be the witnessing functions for Γ in $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. The witnessing functions for $\mathcal{S}$ can be defined as

$$F_j()(i) \leftrightarrow [(\neg\varphi(G()) \wedge F_j^1()(i)) \vee (\varphi(G()) \wedge F_j^2(G())(i))].$$

Other cases: Other cases, including $\wedge$, $\vee$ $\neg$, and structural rules, are trivial. ■

We can generalize further to consider the case where φ is a $g\Sigma_1^B$ formula. This theorem will be needed in the next chapter, when we prove that **VTC**^{0} can interpret the Δ_1^b comprehension rule.

Corollary 3.22 *(*$g\Sigma_1^B$ **Witnessing Theorem***) Suppose that $\mathcal{T}$ is a Σ_0^B theory over the vocabulary $\mathcal{L}$ which is closed under bitgraph expansion in $\mathcal{T}$. Then for each theorem of $\mathcal{T}$ of the form $\forall\overline{x}\forall\overline{Y}\exists Z\varphi(\overline{x},\overline{Y},Z)$, where φ is $g\Sigma_1^B$, there is a string function symbol F in $\mathcal{L}$ such that $\mathcal{T} \vdash \forall\overline{x}\forall\overline{Y}\varphi(\overline{x},\overline{Y},F(\overline{x},\overline{Y}))$.*

Proof: It suffices to show that if $\mathcal{T} \vdash Qx_1 \leq t_1 \exists Z_1 \ldots Qx_m \leq t_m \exists Z_m \theta(\overline{x}, \overline{Z})$, where θ is Σ_0^B, then there are functions $F_1, \ldots, F_m$ in $\mathcal{T}$ such that $\mathcal{T} \vdash Qx_1 \leq t_1 \ldots Qx_m \leq t_m \theta(\overline{x}, F_1(x_1), \ldots, F_m(x_1, \ldots, x_m))$.

The proof of the above claim is by induction on m. The base case and the induction step both follow from Theorem 3.20. ■

3.3.3 Witnessing Theorem for $\mathbf{VTC}^0$

First, we will define a conservative extension of $\mathbf{VTC}^0$ by adding symbols for $\mathbf{TC}^0$ functions, and their defining axioms. Hence, the witnessing theorem for $\mathbf{VTC}^0$ can be stated as a corollary of Theorem 3.20.

The vocabulary of $\mathbf{VTC}^0$ will be extended to a vocabulary $\mathcal{L}_\infty$, and $\mathbf{VTC}^0$ will be extended to a Σ_0^B theory $\overline{\mathbf{VTC}}^0$, so that $\overline{\mathbf{VTC}}^0$ is a conservative extension of $\mathbf{VTC}^0$, and $\mathcal{L}_\infty$ is closed under bitgraph expansion in $\overline{\mathbf{VTC}}^0$ (Definition 3.19). First, we define the number function $numones(X, i)$ as follows.

$$
\begin{aligned}
& numones(X, 0) = 0 \\
X(i) \quad \supset \quad & numones(X, i+1) = numones(X, i) + 1 \\
\neg X(i) \quad \supset \quad & numones(X, i+1) = numones(X, i).
\end{aligned}
$$

Let $\mathcal{L}_0 = \mathcal{L}_A^2$, $\mathcal{L}_1 = \mathcal{L}_A^2 \cup \{numones\}$. For $n \geq 1$, $\mathcal{L}_{n+1}$ is built from $\mathcal{L}_n$ by adding a string function symbol $F_{t,\varphi}$ and a number function symbol $f_{t,\varphi}$ for each pair (t, φ), where $t(\overline{x}, \overline{Y})$ is a $\mathcal{L}_n$ number term, and $\varphi(i, \overline{x}, \overline{Y})$ a $\Sigma_0^B(\mathcal{L}_n)$ formula (all free variables in t and φ have been indicated).

Now, let $\mathcal{L}_\infty = \bigcup_{n \geq 0} \mathcal{L}_n$. Let $\mathcal{T}_\infty$ be the theory obtained from $\mathbf{V}^0$ by replacing Σ_0^B *COMP* by $\Sigma_0^B(\mathcal{L}_\infty)$ *COMP*, the defining axioms for *numones* above, and for each symbols $F_{t,\varphi}$ and $f_{t,\varphi}$, we have the defining axioms

$$
\forall i F_{t,\varphi}(\overline{x}, \overline{Y})(i) \leftrightarrow [i < t \wedge \varphi(i, \overline{x}, \overline{Y})] \quad \text{and} \quad f_{t,\varphi}(\overline{x}, \overline{Y}) = |F_{t,\varphi}(\overline{x}, \overline{Y})|.
$$

It follows from Theorem 3.18, and from the fact that *NUMONES* is provable using $\Sigma_0^B(\mathcal{L}_1)$ *COMP*, that $\mathcal{T}_\infty$ is a conservative extension of $\mathbf{VTC}^0$.

Lemma 3.23 *The theory $\mathcal{T}_\infty$ is conservative over* $\mathbf{VTC}^0$.

Proof: Let $\mathcal{T}_n$ be the extension of $\mathbf{V}^0$, where Σ_0^B *COMP* is replaced by $\Sigma_0^B(\mathcal{L}_n)$ *COMP*, and the defining axioms for symbols in $\mathcal{L}_n$ are added. It is straightforward that the function symbols in $\mathcal{L}_{n+1}$ are Σ_1^B-definable in $\mathcal{T}_n$. Consequently, by Theorem 3.18, $\mathcal{T}_{n+1}$ is conservative over $\mathcal{T}_n$, for $n \geq 0$. ■

It can be seen that the vocabulary $\mathcal{L}_\infty$ is so rich that $\Sigma_0^B(\mathcal{L}_\infty)$ *COMP* is redundant in $\mathcal{T}_\infty$. In fact, let $\overline{\mathbf{VTC}}^0$ be $\mathcal{T}_\infty$ without the axiom scheme $\Sigma_0^B(\mathcal{L}_\infty)$ *COMP*. Then $\overline{\mathbf{VTC}}^0$ is equivalent to $\mathcal{T}_\infty$, and thus conservative over $\mathbf{VTC}^0$.

Next, we show that the function symbols in $\mathcal{L}_\infty$ stand for $\mathbf{TC}^0$ functions.

Lemma 3.24 *The functions in $\mathcal{L}_\infty$ are in* $\mathbf{FTC}^0$.

Proof: The proof is by induction on n that the symbols in $\mathcal{L}_n$ stand for functions in $\mathbf{FTC}^0$. The base case is trivial. The induction step follows from definition of $\mathbf{FTC}^0$ (Definition 2.14), Theorem 2.15, and Corollary 2.16. ■

Now, as a corollary of Theorem 3.20, we have the witnessing theorem for $\mathbf{VTC}^0$. For each function symbol F in $\mathcal{L}_\infty$, let $AX(F)$ be the defining axiom of F in $\overline{\mathbf{VTC}}^0$. Note that this involves only a finite number of function symbols (which are used in defining F), together with their defining axioms.

Corollary 3.25 *(***Witnessing Theorem for $\mathbf{VTC}^0$***) Suppose* $\mathbf{VTC}^0 \vdash \exists Z \varphi(\overline{x}, \overline{Y}, Z)$, *where φ is a Σ_0^B formula. Then there is a string function $F(\overline{x}, \overline{Y})$ in $\mathcal{L}_\infty$ such that* $\mathbf{VTC}^0 + AX(F) \vdash \varphi(\overline{x}, \overline{Y}, F(\overline{x}, \overline{Y}))$. ■

3.4 An Example: Proving Pigeon Hole Principle in $\mathbf{VTC}^0$

We give an example of reasoning in $\mathbf{VTC}^0$, by showing how to prove the Pigeon Hole Principle (*PHP*). The *PHP* states that for any mapping from a set of a numbers to a set of $(a-1)$ numbers, there must be 2 numbers in the domain that have the same image. In the following definition, the mapping is represented as a set of pairs of pre-images and images.

Theorem 3.26 *Let PHP be the following sentence:*

$$\forall a \forall X\ [\forall i \leq a \exists j < a X(i,j)\ \supset\ \exists j < a \exists i_1 \leq a \exists i_2 < i_1 (X(i_1,j) \wedge X(i_2,j))].$$

Then $\mathbf{VTC}^0 \vdash PHP$.

Proving *PHP* involves formalizing a number of concepts, such as set union, total number of bits in an array, etc. These functions will be easily seen to be Σ_1^B-definable in $\mathbf{VTC}^0$. Hence, these symbols can appear in the Σ_0^B *COMP* axiom scheme. In particular, we can apply (number) induction on Σ_0^B formulas in the extended vocabulary. Indeed, we will prove *PHP* in a conservative extension of $\mathbf{VTC}^0$. Recall the definition of *numones* on page 38. Other functions are as follows.

Bounded union of 2 sets: $Union(b,X,Y)(i) \leftrightarrow i < b \wedge (X(i) \vee Y(i))$.

Bounded union of a number of sets (as rows in an array):

$$UnionRows(a,b,Z)(i) \leftrightarrow i < b \wedge \exists j < a Z^{[j]}(i).$$

Total number of bits in an array: $totNumones(0,b,Z) = 0$

$$totNumones(a+1,b,Z) = totNumones(a,b,Z) + numones(Z^{[a]},b).$$

Before proving Theorem 3.26, we prove the following lemmas. First, we prove that the number of bits in the union of two sets is not greater than the sum of the numbers of bits in each set.

Lemma 3.27 $\mathbf{VTC}^0 \vdash \mathit{numones}(\mathit{Union}(b, X, Y), b) \leq \mathit{numones}(X, b) + \mathit{numones}(Y, b)$.

Proof: Trivial by induction on b. ■

Next, we show that the above result also holds when we take the union of the rows in an array.

Lemma 3.28 $\mathbf{VTC}^0 \vdash \mathit{numones}(\mathit{UnionRows}(a, b, Z), b) \leq \mathit{totNumones}(a, b, Z)$.

Proof: The statement is in Σ_0^B. Therefore we can prove it using induction on a. The base case is trivial. Consider the induction step. It is obvious that $\mathit{UnionRows}(a+1, b, Z) = \mathit{Union}(b, \mathit{UnionRows}(a, b, Z), Z^{[a]})$. We have

$$\begin{aligned} \mathit{numones}(\mathit{UnionRows}(a+1, b, Z),\ b) &= \mathit{numones}(\mathit{Union}(b, \mathit{UnionRows}(a, b, Z), Z^{[a]}),\ b) \\ &\leq \mathit{numones}(\mathit{UnionRows}(a, b, Z),\ b) + \mathit{numones}(Z^{[a]},\ b) \\ &\leq \mathit{totNumones}(a, b, Z) + \mathit{numones}(Z^{[a]},\ b) \\ &= \mathit{totNumones}(a+1, b, Z) \end{aligned}$$

■

Lemma 3.29 $\mathbf{VTC}^0 \vdash \forall j < a\ \mathit{numones}(Z^{[j]}, b) \leq k \ \supset\ \mathit{totNumones}(a, b, Z) \leq ak$.

Proof: The lemma is easily proved by induction on a. ■

Proof: (**of Theorem 3.26**) Let Z be the transpose of X, i.e., $\forall j < a \forall i \leq a\ [Z^{[j]}(i) \leftrightarrow X^{[i]}(j)]$. ($Z$ exists by Σ_0^B *COMP*.) The conclusion holds if some rows $Z^{[j]}$ of Z contain at least two values i_1, i_2 (i.e., both $Z^{[j]}(i_1)$ and $Z^{[j]}(i_2)$ hold). We prove this by contradiction. Suppose that $\forall j < a\ \mathit{numones}(Z^{[j]}, a{+}1) \leq 1$. By Lemma 3.29, $\mathit{totNumones}(a, a{+}1, Z) \leq a$. By Lemma 3.28, $\mathit{numones}(\mathit{UnionRows}(a, a+1, Z), a+1) \leq a$. However, it is obvious that $\forall i \leq a\ \mathit{UnionRows}(a, a+1, Z)(i)$. By a simple induction argument, this implies $\mathit{numones}(\mathit{UnionRows}(a, a+1, Z), a+1) = a+1$, a contradiction. ■

Chapter 4

RSUV Isomorphism between $\mathbf{VTC}^0$ and Δ_1^b-CR

In this chapter we will show the RSUV isomorphism between $\mathbf{VTC}^0$ and the first-order theory Δ_1^b-$\mathbf{CR}$ [20]. From this isomorphism and the results in [20], we have the following implication from the possible collapse of $\mathbf{V}^1$ to $\mathbf{VTC}^0$.

Corollary 4.1 *If* $\mathbf{VTC}^0 = \mathbf{V}^1$ *then* $\mathbf{NP}$ *is contained in non-uniform* $\mathbf{TC}^0$*.* ■

From this isomorphism, it also follows that $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$, for some constant i. The reason is that, Δ_1^b-$\mathbf{CR}$ is finitely axiomatizable, since $\mathbf{VTC}^0$ is (Corollary 3.5). Let i be large enough such that proving the finitely many axioms of Δ_1^b-$\mathbf{CR}$ requires at most i applications of the Δ_1^B comprehension rule. Then $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$. This is a positive answer to an open question in [20].

Corollary 4.2 *For some constant* i*,* $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$*.* ■

Proving this isomorphism consists of interpreting $\mathbf{VTC}^0$ in Δ_1^b-$\mathbf{CR}$ and vice versa. Interpreting $\mathbf{VTC}^0$ in Δ_1^b-$\mathbf{CR}$ is straightforward, while the other direction is more complicated, involving defining the multiplication function for strings, proving its basic properties; and interpreting *BASIC* axioms, the axiom scheme *Open-LIND*, and the Δ_1^B comprehension rule. In the following sections, we will first present the theory Δ_1^b-$\mathbf{CR}$, then show a bi-interpretation between $\mathbf{VTC}^0$ and Δ_1^b-$\mathbf{CR}$.

4.1 The Theory Δ_1^b-CR

First, recall that the language of Bounded Arithmetic [4] is

$$\mathcal{L}_{BA} = [0, S, +, \cdot, \lfloor \tfrac{1}{2}x \rfloor, |x|, x\#y; \leq].$$

The intended meaning of the symbols are as follows: S is the successor function, $|x|$ is the length of the binary representation of x, $x\#y = 2^{|x||y|}$, and other symbols have standard meanings. The language $\mathcal{L}_{\Delta_1^b CR}$ of Δ_1^b-$\mathbf{CR}$ is $\mathcal{L}_{BA}$ together with $\dot{-}$ and MSP as in [25]. Here, $MSP(x, i)$ is the most significant part of x, ignoring the last i bits: $MSP(x, i) = \lfloor x/2^i \rfloor$; and if $x < y$ then $x \dot{-} y = 0$, otherwise, $x \dot{-} y = x - y$. In defining Δ_1^b-$\mathbf{CR}$, the binary predicate Bit is also used. Here $Bit(i, x) = 1$ if and only if the ith bit in the binary representation of x is 1. [1]

The set $BASIC$ of basic axioms for function symbols in $\mathcal{L}_{\Delta_1^b CR}$ consists of 32 axioms for function symbols in $\mathcal{L}_{BA}$ (also called $BASIC$ in [4]), together with the defining axioms for the new symbols $\dot{-}$ and MSP. The theory Δ_1^b-$\mathbf{CR}$ is axiomatized by $BASIC$, the axiom scheme *Open-LIND*, and the Δ_1^b bit-comprehension rule (the last two will be defined below).

The axiom scheme *Open-LIND* can be seen as a scheme of induction on "small" numbers (i.e., $|z|$) for quantifier-free formulas. Formally, *Open-LIND* is

$$[\varphi(0) \wedge \forall x(\varphi(x) \supset \varphi(Sx))] \supset \forall z \varphi(|z|), \tag{4.1}$$

where φ is an open formula. The Δ_1^b bit-comprehension rule is defined as follows. First, given a formula φ, the comprehension axiom for it, denoted by $COMP_\varphi$, is the formula

$$\exists x < 2^{|a|} \forall i < |a| [Bit(i, x) \leftrightarrow \varphi(i)].$$

[1]Note that in [20], the order of the arguments of Bit is different. Here, we follow the convention in [12].

Then, the Δ_1^b bit-comprehension rule is the following inference rule:

$$\frac{\varphi(i) \leftrightarrow \psi(i)}{COMP_\varphi(t)}$$

where φ is a Σ_1^b formula, and ψ is a Π_1^b formula. Note that there is no side formula in the rule, and thus it is apparently weaker than the Δ_1^b comprehension axiom scheme:

$$\forall i(\varphi(i) \leftrightarrow \psi(i)) \supset COMP_\varphi(t)$$

(where φ, ψ are Σ_1^b formulas). Note also that here, Σ_1^b and Π_1^b formulas can be translated to $g\Sigma_1^B$ and $g\Pi_1^B$ formulas, respectively.

4.2 Interpreting $\mathbf{VTC}^0$ in Δ_1^b-$\mathbf{CR}$

We will show that formulas of $\mathbf{VTC}^0$ can be translated into formulas of Δ_1^b-$\mathbf{CR}$, such that theorems of $\mathbf{VTC}^0$ are translated into theorems of Δ_1^b-$\mathbf{CR}$. This can be done using the model-theoretic approach as follows [22, Section 5.5]. Given a model $\mathcal{N}$ of Δ_1^b-$\mathbf{CR}$, let $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$, where $\mathcal{M}_2 = \mathcal{N}$ and $\mathcal{M}_1 = \{|x| : x \in \mathcal{N}\}$. Moreover, in $\mathcal{M}$, the constants and function symbols $0, 1, +, \cdot, |\,|$ are interpreted as in $\mathcal{N}$, where now $+, \cdot$ are restricted to $\mathcal{M}_1$. The predicate $\in$ is interpreted as $x \in Y$ iff $Bit(x, Y) = 1$. Then, it suffices to show that $\mathcal{M}$ is a model of $\mathbf{VTC}^0$.

First, it is straightforward to show that the axioms **B1**-**B14**, **L1**-**L2**, and **SE** (page 25) hold in $\mathcal{M}$. Second, the Σ_0^B comprehension scheme holds in $\mathcal{M}$ because Δ_1^b bit-comprehension rule holds in $\mathcal{N}$. Next, the axiom *NUMONES* holds in $\mathcal{M}$ as seen from a proof of how to define *numones* from multiplication. In particular, suppose that $x_{n-1} \ldots x_0$ is the binary representation of x. Let $x' = x_{n-1}0 \ldots 0x_{n-2}0 \ldots 0x_0$, where every block of 0's has length $(1 + |n|)$. Let $z = 10 \ldots 010 \ldots 01$ (n 1's, and each block of 0's has length $1 + |n|$). It is straightforward that the counting array for x can be extracted from the product $x'.z$.

4.3 Interpreting Δ_1^b-CR in $\mathbf{VTC}^0$

Let $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ be a model of $\mathbf{VTC}^0$. We show that Δ_1^b-**CR** is interpretable in $\mathcal{M}$ by showing that the structure $\mathcal{N}$, whose universe is $\mathcal{M}_2$, is a model of Δ_1^b-**CR**. This involves showing that the symbols of $\mathcal{L}_{\Delta_1^b CR}$ can be interpreted in $\mathcal{N}$ (so the *BASIC* axioms are satisfied), and that the axiom *Open-LIND* as well as the Δ_1^b bit-comprehension rule hold in $\mathcal{N}$.

Recall the *Open-LIND* axiom scheme in Equation 4.1. It is straightforward to show that it is equivalent to the following axiom

$$[\varphi(0) \wedge \forall x \leq |z|(\varphi(x) \supset \varphi(Sx))] \supset \varphi(|z|),$$

for open formula φ. A direct translation of the latter is the Σ_0^B number induction scheme (page 26), which is provable in $\mathbf{V}^0$.

For defining symbols of $\mathcal{L}_{\Delta_1^b CR}$, observe that the theory $\mathbf{A}^1$ defined in [16] is equivalent to $\mathbf{V}^1$. Bi-interpretation between $\mathbf{A}^1$ and S_2^1 has been carried out in detail in [16]. In particular, interpretation of functions symbols of S_2^1 in $\mathbf{A}^1$ can be found there. Although $\mathbf{A}^1$ (equivalently $\mathbf{V}^1$) is apparently more powerful than $\mathbf{VTC}^0$, much of this interpretation can be done in $\mathbf{VTC}^0$. On the other hand, there are techniques employed in [16] that do not seem to apply for $\mathbf{VTC}^0$. In particular, the proof of associativity of addition, and the way iterated addition is defined, require some modification. Note that the remaining symbols in $\mathcal{L}_{\Delta_1^b CR}$ (i.e., $\dot{-}$, *MSP*, *Bit*) can be easily defined in $\mathbf{VTC}^0$ (in fact, in $\mathbf{V}^0$).

In the following parts, we will first highlight the necessary modifications of the interpretation of $\mathcal{L}_{BA}$ functions symbols from [16]; these include proving addition associativity (we actually show that it holds in $\mathbf{V}^0$), defining multiplication for strings, and proving the distributive laws. Then we will show that $\mathbf{VTC}^0$ admits the Δ_1^b bit-comprehension rule.

4.3.1 Proving Addition Associativity in $\mathbf{V}^0$

String addition is defined using the conventional method. Suppose $\varphi_+(i, X, Y)$ represents the carry at bit position i when adding X and Y. Then the ith bit of the sum $X + Y$ is exactly

$$X(i) \oplus Y(i) \oplus \varphi_+(i, X, Y).$$

Definition 4.3 *(***String Addition***) Let $\varphi_+(i, X, Y)$ be the Σ_0^B formula*

$$\exists j < i[X(j) \wedge Y(j) \wedge \forall l < i(j < l \supset X(l) \oplus Y(l))]. \tag{4.2}$$

Then string addition is defined as follows:

$$(X + Y)(i) \leftrightarrow [(i < |X| + |Y|) \wedge (X(i) \oplus Y(i) \oplus \varphi_+(i, X, Y))].$$

By this definition, string addition is Σ_1^B-definable in $\mathbf{V}^0$. Moreover, we will show that $\mathbf{V}^0$ is powerful enough for proving the associativity of string addition.

Lemma 4.4 *(***Associativity of String Addition***)* $\mathbf{V}^0$ *proves the associativity of string addition, i.e.,*

$$\mathbf{V}^0 \vdash X + (Y + Z) = (X + Y) + Z.$$

Proof: We will show that the ith bits of LHS and RHS are the same, i.e.,

$$(X + (Y + Z))(i) \leftrightarrow ((X + Y) + Z)(i).$$

This is equivalent to (recall that $\varphi_+(i, X, Y)$ expresses the carry bit, cf. Equation 4.2)

$$X(i) \oplus (Y + Z)(i) \oplus \varphi_+(i, X, Y + Z) \leftrightarrow (X + Y)(i) \oplus Z(i) \oplus \varphi_+(i, X + Y, Z). \tag{4.3}$$

By expanding $(Y + Z)(i)$ on the LHS, and $(X + Y)(i)$ on the RHS, we can see that the $X(i), Y(i), Z(i)$ components on both sides are the same. Thus, we can simplify Equation 4.3. Formally,

$$(Y + Z)(i) \leftrightarrow Y(i) \oplus Z(i) \oplus \varphi_+(i, Y, Z),$$

$$(X + Y)(i) \leftrightarrow X(i) \oplus Y(i) \oplus \varphi_+(i, X, Y).$$

Therefore, Equation 4.3 is equivalent to

$$\varphi_+(i,Y,Z) \oplus \varphi_+(i,X,Y+Z) \leftrightarrow \varphi_+(i,X,Y) \oplus \varphi_+(i,X+Y,Z).$$

In order to prove the above statement, we will prove a stronger result. Let a_i, b_i, c_i and d_i denote $\varphi_+(i,Y,Z)$, $\varphi_+(i,X,Y+Z)$, $\varphi_+(i,X,Y)$ and $\varphi_+(i,X+Y,Z)$ respectively. We will show by induction on i that

$$(a_i \wedge b_i \leftrightarrow c_i \wedge d_i) \wedge (a_i \vee b_i \leftrightarrow c_i \vee d_i).$$

The base case is trivial, since

$$V^0 \vdash \neg a_0 \wedge \neg b_0 \wedge \neg c_0 \wedge \neg d_0.$$

The induction step follows from the recursive evaluation of a_i b_i, c_i and d_i:

$$\begin{aligned}
a_{i+1} &= [Y(i) \wedge Z(i)] \vee [(Y(i) \oplus Z(i)) \wedge a_i], \\
b_{i+1} &= [X(i) \wedge (Y(i) \oplus Z(i) \oplus a_i)] \vee [(X(i) \oplus Y(i) \oplus Z(i) \oplus a_i) \wedge b_i], \\
c_{i+1} &= [X(i) \wedge Y(i)] \vee [(X(i) \oplus Y(i)) \wedge c_i], \\
d_{i+1} &= [Z(i) \wedge (X(i) \oplus Y(i) \oplus c_i)] \vee [(X(i) \oplus Y(i) \oplus Z(i) \oplus c_i) \wedge d_i].
\end{aligned}$$

■

4.3.2 Defining Multiplication in $\mathbf{VTC}^0$

We will now define multiplication for the strings of $\mathcal{M}$, and prove its basic properties. In [16], a definition of string multiplication is given in such a way that commutativity follows immediately. In this definition, first, a matrix $X \times Y$ is defined symmetrically with respect to the two arguments X, Y (i.e., $X \times Y = Y \times X$). Then, the product $X \cdot Y$ is obtained by iteratively adding the rows of $X \times Y$. It can be seen that the second step is possible thanks to Σ_1^B number induction (page 26), which might not be in $\mathbf{VTC}^0$. Therefore another method is required. From the complexity point of view, adding n strings, each of length m, is possible in $\mathbf{TC}^0$ [9, 2, 3]. We will show that there is indeed an uniform way of doing this, i.e., it can be carried out in the theory $\mathbf{VTC}^0$. The method that we use is adapted from [3], and is based on the fact that counting the number of bits in a string is available in $\mathbf{VTC}^0$.

In the following parts, we will formalize various concepts in $\mathbf{VTC}^0$ by introducing symbols for functions that are Σ_1^B-definable in $\mathbf{VTC}^0$. More precisely, suppose that we extend $\mathbf{VTC}^0$ by adding these new symbols for these functions, together with their defining axioms, then the new theory is conservative over $\mathbf{VTC}^0$. We will therefore tacitly assume that the language of $\mathbf{VTC}^0$ contains these symbols, together with their defining axioms.

Adding n Strings

First, we will show that simultaneously counting numbers of bits in every column of a matrix is possible in $\mathbf{VTC}^0$. Formally, suppose that X is a matrix of size $n \times m$ (n rows, each of length $\leq m$), let X' be the transpose of X. Then $\mathbf{VTC}^0$ proves the existence of Z, where for every $i < m$, the unique j such that $Z^{[i]}(j)$ is the number of bits in $X'^{[i]}$:

$$\mathbf{VTC}^0 \vdash \exists Z \leq \langle m, n \rangle \forall i < m \forall j \leq n [Z^{[i]}(j) \leftrightarrow j = numones(X'^{[i]}, n)]. \tag{4.4}$$

Thus, in $\mathbf{VTC}^0$ we can define the string function $SimulCountCols(n, m, X)$, where the ith row of $SimulCountCols(n, m, X)$ contains only the number of bits in column i of X (i.e., $numones(X'^{[i]}, n)$), for $i < m$. Now, the sum of n rows of an array X will be defined as a function of $SimulCountCols(n, m, X)$ (where m is the bound on the lengths of the rows in X). We will need the number functions $\lceil x/y \rceil$, $|x|$ and 2^y (for $y \leq |a|$), and the relation $Bit(i, x)$ (page 43). These functions and relation are definable in $I\Delta_0$, and thus also in $\mathbf{V}^0$ [12].

Suppose that we have n strings with length bounded by m, which are represented as the rows of a matrix X. Let $Z = SimulCountCols(n, m, X)$. Let $first(X)$ be the smallest

z such that $X(z)$ holds. (We will often use $\mathit{first}(X)$ to get the unique z such that $X(z)$ holds.) For convenience, let c_i be the unique z such that $Z^{[i]}(z)$, i.e., $c_i = \mathit{first}(Z^{[i]})$. Then $c_i \leq n$, for $0 \leq i < m$. Intuitively,

$$\sum_{i=0}^{n-1} X^{[i]} = \sum_{i=0}^{m-1} 2^i c_i,$$

where the RHS should be seen as a string S of length $\leq m + |n|$. We are going to show the existence of such string S.

We will exhibit two strings L and H such that $S = L + H$. First, divide the sequence $c_{m-1}, \ldots, c_0$ into $2k$ blocks, each of length l (l, k will be made precise later). Suppose that the blocks are numbered with $0, \ldots, (2k-1)$, i.e., the block i is $c_{(i+1)l-1}, \ldots, c_{il}$. The number l will be small enough so that for each block i, the following sum is a number $< 2^{2l}$:

$$b_i = \sum_{j=0}^{l-1} 2^j c_{il+j}. \tag{4.5}$$

Here, the LHS should be seen as a number function of Z, il, and l, i.e., $b_i = f(Z, il, l)$ for some number function f. Then, informally, L is the sum of the blocks $0, 2, \ldots$, and H is the sum of the other blocks, i.e.,

$$L = \sum_{i=0}^{k-1} 2^{2il} b_{2i}, \quad H = \sum_{i=0}^{k-1} 2^{(2i+1)l} b_{2i+1}.$$

Since $b_i < 2^{2l}$, L is simply the the concatenation of $b_0, b_2, \ldots$, and similarly, H is the concatenation of $b_1, b_3, \ldots$ (we may need to pad b_i's with the right number of 0's). Thus, L, H are Σ_1^B-definable in $\mathbf{VTC}^0$.

Formally, we define a number function $f(Z, i, l)$ so that in Equation 4.5, $b_i = f(Z, il, l)$. This function is defined only when $l < |a|$, for some a. We need

$$f(X, i, l) = \sum_{j=0}^{l-1} 2^j c_j.$$

The value of $f(X, i, l)$ is the number of bits in a "long" string Y, which contains: 1 substring of c_i 1's; 2 substrings, each of c_{i+1} 1's, ..., 2^{l-1} substrings, each of c_{i+l-1} 1's.

Constructing the string Y is obvious. Note that the bound on c_j can also taken to be $b = |X|$. Let Y be such that

$$\forall j < l \forall p < 2^j \forall q < b \, [Y((2^j - 1)b + pb + q) \leftrightarrow q < c_j].$$

Then $f(Z, i, l)$ is defined as

$$f(Z, i, j) = numones(Y, 2^l b).$$

It is easy to prove the basic properties of f:

$$f(Z, i, 0) = \mathit{first}(Z^{[i]}), \quad f(Z, i, l+1) = f(Z, i, l) + 2^l \mathit{first}(Z^{[i+l]}), \quad f(Z, i, l) < b2^l,$$

where b is such that $\mathit{first}(Z^{[j]}) < b$ for all j.

Now, let $l = 1 + |n|$, $k = \lceil m/2l \rceil$, $b_i = f(Z, il, l)$. Then L and H can be defined using Σ_0^B comprehension as follows:

$$\forall i < k \forall j < 2l[L(2il + j) \leftrightarrow Bit(j, b_{2i})], \quad \forall i < k \forall j < 2l[H((2i+1)l + j) \leftrightarrow Bit(j, b_{2i+1})].$$

Let $Accumulate(n, m, Z)$ denote the value of the string S that we have just proved to exist. (Thus, intuitively,

$$Accumulate(n, m, Z) = \sum_{i=0}^{m-1} c_i 2^i,$$

where c_i is the unique number present in $Z^{[i]}$, and $c_i \leq n$ for $0 \leq i < m$.) It follows that $Accumulate$ is Σ_1^B-definable in $\mathbf{VTC}^0$. Now, let $Sum(n, m, X)$ denote the sum of the first n rows of X (where the length of each row is bounded by m). Then $Sum(n, m, X) = Accumulate(n, m, Z)$, i.e.,

$$Sum(n, m, X) = Accumulate(n, m, SimulCountCols(n, m, X)).$$

Therefore, $Sum(n, m, X)$ is Σ_1^B-definable in $\mathbf{VTC}^0$.

A Symmetric Definition of Multiplication

The conventional algorithm for multiplying X and Y is to write down a matrix $X \otimes Y$, where each row corresponds to a bit in Y, and is either 0 or a copy of X padded with the right number of 0's. Then, the product $X \cdot Y$ is the sum of the rows of this matrix. The disadvantage of this definition is that it seems difficult (in $\mathbf{VTC}^0$) to prove that $X \cdot Y = Y \cdot X$.

We will modify this algorithm by using a different matrix $X \times Y$, which is symmetric with respect to X and Y (i.e., $X \times Y = Y \times X$), and which gives the same results in the standard structure $\underline{\mathbb{N}}_2$. In the usual definition, when $X(i) = Y(j) = 1$, a bit is present at position $i + j$ in row j of the matrix $X \otimes Y$ (i.e., we record 2^{i+j} on row j of $X \otimes Y$). For $X \times Y$, instead of recording 2^{i+j} on row j, we record it on row ij. The only problem is the duplication in the case where $i \neq j$ and both $X(i) = Y(j) = 1$ and $X(j) = Y(i) = 1$. In this case, we also record 2^{i+j} on the row $|X||Y| + ij$. It is obvious that $X \times Y = Y \times X$, and that $X \times Y$ has $2|X||Y|$ rows, each of length $|X| + |Y|$. The product $X \cdot Y$ is the sum of the rows of $X \times Y$.

Definition 4.5 *(***String Multiplication***) Let $X \times Y$ be defined as*

$$(X \times Y)(k) \leftrightarrow [k \leq \langle 2|X||Y|, |X| + |Y| \rangle] \wedge \exists i < |X| \exists j < |Y|$$
$$[X(i) \wedge Y(j) \wedge [k = \langle ij, i+j \rangle \vee (i \neq j \wedge X(j) \wedge Y(i) \wedge k = \langle ij + |X||Y|, i+j \rangle)]].$$

Then the product of strings X, Y is

$$X \cdot Y = Sum(2|X||Y|, |X| + |Y|, X \times Y).$$

It is easy to show that string multiplication defined in this way is commutative.

Lemma 4.6 $\mathbf{VTC}^0 \vdash X \times Y = Y \times X, \quad \mathbf{VTC}^0 \vdash X \cdot Y = Y \cdot X.$ ■

4.3.3 Proving the Distributive Law

We need to prove the following lemma:

Lemma 4.7 *(***Distributive Law***)* $\mathbf{VTC}^0 \vdash X \cdot (Y + Z) = X \cdot Y + X \cdot Z$.

First, let $\{a\}$ denote the set of one element a. We will first prove a special case of Lemma 4.7, from which the lemma will follow by a simple induction argument. (Note that when $|X| < a$, $X + \{a\}$ is just the union of X and $\{a\}$.)

Claim 4.8 $\mathbf{VTC}^0 \vdash |X| < a \supset (X + \{a\}) \cdot Y = X \cdot Y + \{a\} \cdot Y$.

Essentially, proving this claim amounts to showing that

$$Sum(n, m, Z) + Z^{[n]} = Sum(n + 1, m, Z).$$

This is similar to Lemma 7.9 in [3]. However, the proof in [3] does not consider the case where $|n + 1| = 1 + |n|$, i.e., the sizes of the blocks used in defining $Accumulate(n, m, Z)$ and $Accumulate(n + 1, m, Z)$ are different. We will show that in $\mathbf{VTC}^0$, different values can be used for the block size (as long as they are $\geq 1 + |n|$). In other words, we will show that

$$Accumulate(n, m, Z) = Accumulate(n', m, Z), \tag{4.6}$$

for $n' > n$. Then the approach used in [3] can be formalized in $\mathbf{VTC}^0$, but we can also use a simpler method.

Let $ToString(k, x)$ be the string obtained by padding k 0's to the end of the binary representation of x:

$$\forall i \leq (k + |x|)[ToString(k, x)(i) \leftrightarrow (k \leq i \wedge Bit(i - k, x))].$$

Recall (page 48) that $first(X)$ is the smallest value of z such that $X(z)$ holds. Assume $first(Z^{[j]}) \leq n$, for all $j < m$. Then, intuitively,

$$Accumulate(n, m, Z) = \sum_{j=0}^{m-1} ToString(j, first(Z^{[j]})).$$

Formally, it is straightforward to derive the following recursion for $Accumulate$:

$$Accumulate(n, m + 1, Z) = Accumulate(n, m, Z) + ToString(m, first(Z^{[m]})).$$

This recursion does not depend on n. Thus, Equation 4.6 follows.

Proof: (**of Claim 4.8**)

As we have shown above, we can use block size $l = |n'|$ in the definition of *Accumulate*, for any $n' \geq n$. Thus, we can use the same proof as that of Lemma 7.9 in [3]. For another proof, let Z, Z_1 be as in the definition of $(X + \{a\}) \cdot Y$ and $X \cdot Y$, i.e.,

$$\begin{aligned} Z &= \mathit{SimulCountCols}(2a|Y|, a + |Y|, (X + \{a\}) \times Y), \\ Z_1 &= \mathit{SimulCountCols}(2|X||Y|, |X| + |Y|, X \times Y). \end{aligned}$$

Then, informally, the difference between Z and Z_1 is exactly $\{a\} \cdot Y$. More precisely, for $j < a + |Y|$,

$$\begin{aligned} \mathit{first}(Z^{[j]}) = \mathit{first}(Z_1^{[j]}) \quad &\text{iff } j < a \text{ or } \neg Y(j - a), \\ \mathit{first}(Z^{[j]}) = \mathit{first}(Z_1^{[j]}) + 1 \quad &\text{iff } j \geq a \text{ and } Y(j - a). \end{aligned}$$

Thus, by induction on $i < a + |Y|$, we can show that

$$\mathit{Accumulate}(n, i, Z) = \mathit{Accumulate}(n, i, Z_1) + \mathit{Chop}(\{a\} \cdot Y, i),$$

where $n = 2a|Y|$, and $\mathit{Chop}(X, i)$ is the last i bits of X, i.e., $\mathit{Chop}(X, i)(j) \leftrightarrow j < i \wedge X(j)$.

■

The proof of Lemma 4.7 now follows.

Proof: (**of Lemma 4.7**)

We can prove by induction on i that

$$|X| \leq i \supset [X \cdot (Y + Z) = X \cdot Y + X \cdot Z].$$

The base case is trivial. The induction step follows from Claim 4.8. ■

4.3.4 Interpreting the Δ_1^b Comprehension Rule in $\mathbf{VTC}^0$

The $g\Delta_1^B$ comprehension rule in second-order setting is a direct translation of the Δ_1^b comprehension rule in first-order logic. (Note that Σ_1^B and Π_1^B correspond to strict Σ_1^b

and strict Π_1^b, while $g\Sigma_1^B$ and $g\Pi_1^B$ correspond to Σ_1^b and Π_1^b, respectively.) It can be stated formally as follows.

Definition 4.9 *(*$g\Delta_1^B$ **Comprehension Rule***) A theory T is said to admit the $g\Delta_1^B$ comprehension rule if whenever*

$$T \vdash \forall i < b[\varphi(i) \leftrightarrow \psi(i)],$$

for some $g\Sigma_1^B$ formulas φ and $g\Pi_1^B$ formula ψ, then

$$T \vdash \exists X \leq b \forall i < b[X(i) \leftrightarrow \varphi(i)].$$

This rule is apparently weaker than the $g\Delta_1^B$ comprehension axiom. In particular, $\mathbf{VTC}^0$ may not prove the axiom, but, we will show that it admits the rule. It suffices to show that $\mathbf{VTC}^0$ admits the $g\Sigma_1^B$ replacement rule, which is defined as follows.

Definition 4.10 *(*$g\Sigma_1^B$ **Replacement Rule***) A theory T is said to admit the $g\Sigma_1^B$ replacement rule if whenever*

$$T \vdash \forall i < b \exists Z < b\varphi(i, Z),$$

for some $g\Sigma_1^B$ formula φ, which may contain other free variables, then

$$T \vdash \exists W < \langle b, b\rangle \forall i < b\varphi(i, W^{[i]}).$$

Lemma 4.11 *The theory $\mathbf{VTC}^0$ admits the $g\Sigma_1^B$ replacement rule.*

Proof: The proof is a straightforward application of the $g\Sigma_1^B$ Witnessing Theorem (Corollary 3.22), applied for $\mathbf{VTC}^0$. The idea is to use the witnessing function for $\exists Z < b\varphi(i, Z)$ (where φ is a $g\Sigma_1^B$ formula) to construct the witnessing function for $\exists W < \langle b, b\rangle \forall i < b\varphi(i, W^{[i]})$. Formally, let $\varphi(i, Z)$ be a $g\Sigma_1^B$ formula, which may contain other free variables, and suppose that

$$\mathbf{VTC}^0 \vdash \forall i < b \exists Z < b\varphi(i, Z).$$

Then by Corollary 3.22, there is a $\mathbf{TC}^0$-function $F(i)$ such that $|F(i)| < b$ for $i < b$, and

$$\mathbf{VTC}^0 + AX(F) \vdash \forall i < b\varphi(i, F(i)).$$

Let G be the function defined as

$$G()(i,j) \Leftrightarrow i < b \wedge j < b \wedge F(i)(j),$$

then G is in $\mathbf{FTC}^0$. We have

$$\mathbf{VTC}^0 + AX(G) \vdash \forall i < b\varphi(i, G()^{[i]}),$$

hence

$$\mathbf{VTC}^0 + AX(G) \vdash \exists W < \langle b, b\rangle \forall i < b\varphi(i, W^{[i]}).$$

Since $\mathbf{VTC}^0 + AX(G)$ is conservative over $\mathbf{VTC}^0$, the conclusion follows. ■

Corollary 4.12 $\mathbf{VTC}^0$ *admits the* $g\Delta_1^B$ *comprehension rule.*

Proof: It is easy to show that if a theory T admits the $g\Sigma_1^B$ replacement rule, then it also admits the $g\Delta_1^B$ comprehension rule. ■

Chapter 5

Conclusion

We introduce the second-order theory $\mathbf{VTC}^0$, which is finitely axiomatizable and characterizes precisely the complexity class $\mathbf{TC}^0$. We show that it is RSUV isomorphic to the first-order theory Δ_1^b-$\mathbf{CR}$, whose definition requires an inference rule. Thus, we are able to translate into the second-order setting Johannsen and Pollett's result [20] relating the possible collapse of $\mathbf{NP}$ to non-uniform $\mathbf{TC}^0$ and the possible collapse of the corresponding theories. We are also able to answer positively an open question posed in [20], i.e., $\Delta_1^b - \mathbf{CR} = \Delta_1^b - \mathbf{CR}_i$, for some constant i.

We also give an example of reasoning in $\mathbf{VTC}^0$ by proving the pigeonhole principle in $\mathbf{VTC}^0$: *PHP* can be formalized as a Σ_0^B theorem of $\mathbf{VTC}^0$. As noted in [23], $\Sigma_0^{1,b}$ theorems of $(I\Sigma_0^{1,b})^{count}$ can be translated into families of tautologies which have polynomial-size $\mathbf{FC}$ proofs. It can be checked that the same arguments can be carried over to that of $\mathbf{VTC}^0$. In particular, Σ_0^B theorems of $\mathbf{VTC}^0$ can be translated into families of tautologies which have polynomial-size $\mathbf{FC}$ (or equivalently, $\mathbf{TC}^0$-Frege [7]). Consequently, the family of tautologies corresponding to *PHP* has polynomial-size $\mathbf{TC}^0$-Frege proofs. It is plausible that the same is true for the Hex tautologies [5], i.e., the Hex tautologies can be formalized as a Σ_0^B theorem of $\mathbf{VTC}^0$. It follows that the family of tautologies translated from this Σ_0^B theorem has polynomial-size $\mathbf{TC}^0$-Frege proofs.

While Σ_0^B theorems of $\mathbf{VTC}^0$ can be translated as discussed above, we know of no such translation for Σ_1^B theorems of $\mathbf{VTC}^0$ (or equivalent theories). Notice that $\mathbf{VTC}^0$'s Σ_1^B theorems seem to reflect more clearly the class $\mathbf{TC}^0$ than its Σ_0^B theorems, e.g., the $\mathbf{TC}^0$ functions are exactly those Σ_1^B definable in $\mathbf{VTC}^0$. This suggests a possible topic for future research.

It is possible, using a similar approach to the one that we have used here, to obtain "minimal" theories for other complexity classes, such as $\mathbf{ACC}^0[m]$, $\mathbf{NC}^1$, $\mathbf{P}$. The theories for classes of higher complexity than $\mathbf{TC}^0$ certainly contain $\mathbf{VTC}^0$. Proving RSUV isomorphism of these theories with their first-order counterparts (e.g., the first-order theory **AID** for class $\mathbf{NC}^1$ [1]) will therefore be relieved from defining string multiplication and proving its properties. Proving witnessing theorems for these theories may also benefit from our general witnessing theorem.

Bibliography

[1] Toshiyasu Arai. Bounded arithmetic AID for Frege system. Manuscript, 1991.

[2] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On Uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[3] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On Interpolation and Automatization for Frege Systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[4] Samuel Buss. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[5] Samuel Buss. Polynomial Size Proofs of the Propositional Pigeonhole Principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[6] Samuel Buss. The Graph of Multiplication is Equivalent to Counting. *Information Processing Letters*, 41:199–201, 1992.

[7] Samuel Buss and Peter Clote. Cutting Planes, Connectivity and Threshold Logic. *Archive for Mathematical Logic*, 35:33–62, 1996.

[8] Samuel R. Buss, editor. *Handbook of Proof Theory*. Elsevier, Amsterdam, 1998.

[9] Ashok K. Chandra, Larry Stockmeyer, and Uzi Vishkin. Constant Depth Reducibility. *SIAM Journal on Computing*, 13(2):423–439, 1984.

[10] Peter Clote and Evangelos Kranakis. *Boolean Functions and Computation Models*. Springer, 2002.

[11] Peter Clote and Gaisi Takeuti. First Order Bounded Arithmetic and Small Boolean Circuit Complexity Classes. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II*. Birkhäuser, 1995.

[12] Stephen Cook. Proof Complexity and Bounded Arithmetic. Course Notes for CSC 2429S. Department of Computer Science, University of Toronto. Available at http://www.cs.toronto.edu/ sacook/, 2002.

[13] Stephen Cook and Antonina Kolokolova. A Second-order System for Polytime Reasoning Based on Gradel's Theorem. In *In Proceedings Sixteenth Annual IEEE Symposium on Logic in Computer Science (LICS '01)*, pages 177–186, 2001.

[14] Stephen Cook and Neil Thapen. The Strength of Replacement in Weak Arithmetic. Manuscript., 2003.

[15] William Hesse. Division is in Uniform $\mathbf{TC}^0$. In *Eighth International Colloquium on Automata, Languages and Programming (ICALP 2001)*, 2001.

[16] Aleksander Ignjatovic and Phuong Nguyen. Characterizing Polynomial Time Computable Functions Using Theories with Weak Set Existence Principles. In *Computing: The Australasian Theory Symposium*, 2003.

[17] Neil Immerman. *Descriptive Complexity*. Springer, 1999.

[18] Jan Krajíček and Pavel Pudlák and Gaisi Takeuti. Bounded Arithmetic and the Polynomial Hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.

[19] Jan Johannsen. A Bounded Arithmetic Theory for Constant Depth Threshold Circuits. In Petr Hájek, editor, *GÖDEL '96. Springer Lecture Notes in Logic 6*, 1996.

[20] Jan Johannsen and Chris Pollett. On the Δ_1^b-Bit-Comprehension Rule. In Sam Buss, Petr Hájek and Pavel Pudlák, editor, *Logic Colloquium 98*.

[21] Jan Johannsen and Chris Pollett. On Proofs about Threshold Circuits and Counting Hierarchies. In *Proc. 13th IEEE Symposium on Logic in Computer Science*, pages 444–452, 1998.

[22] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, 1995.

[23] Jan Krajíček. On Frege and Extended Frege Proof Systems. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II.* Birkhäuser, 1995.

[24] Alexander A. Razborov. An Equivalence between Second Order Bounded Domain Bounded Arithmetic and First Order Bounded Arithmetic. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 247–277. Oxford, 1993.

[25] Gaisi Takeuti. RSUV Isomorphism. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–386. Oxford, 1930.

[26] Domenico Zambella. Notes on Polynomially Bounded Arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.